Phase 2.0

User Manual

Schrödinger Press

Revision A, April 2006

Contents

Document Conventions

In addition to the use of italics for names of documents, the font conventions that are used in this document are summarized in the table below.

Table 3.1.

Font	Example	Use
Sans serif	Project Table	Names of GUI features, such as panels, menus, menu items, buttons, and labels
Monospace	`$SCHRODINGER/maestro`	File names, directory names, commands, environment variables, and screen output
Italic	*filename*	Text that the user must replace with a value
Sans serif uppercase	CTRL+H	Keyboard keys

In descriptions of command syntax, the following UNIX conventions are used: braces `{ }` enclose a choice of required items, square brackets `[ ]` enclose optional items, and the bar symbol `|` separates items in a list from which one item must be chosen. Lines of command syntax that wrap should be interpreted as a single command.

In this document, to *type* text means to type the required text in the specified location, and to *enter* text means to type the required text, then press the ENTER key.

References to literature sources are given in square brackets, like this: [10].

Introduction to Phase

Phase is a versatile product for pharmacophore perception, structure alignment, activity prediction, and 3D database searching. Given a set of molecules with high affinity for a particular protein target, Phase uses fine-grained conformational sampling and a range of scoring techniques to identify common pharmacophore hypotheses, which convey characteristics of 3D chemical structures that are purported to be critical for binding. Each hypothesis is accompanied by a set of aligned conformations that suggest the relative manner in which the molecules are likely to bind.

A given hypothesis may be combined with known activity data to create a 3D QSAR model that identifies overall aspects of molecular structure that govern activity. This model may be used in conjunction with the hypothesis to mine a 3D database for molecules that are most likely to exhibit strong activity toward the target.

Phase provides support for lead discovery, SAR development, lead optimization and lead expansion. Phase may also be used as a source of molecular alignments for third-party 3D QSAR programs.

Phase is integrated into Maestro, the graphical user interface (GUI) for all Schrödinger products. An introduction to the general capabilities of Maestro is given in Chapter 2. For more detailed information on Maestro, see the Maestro online help or the *Maestro User Manual*. An overview of the Phase interface is given in Chapter 3.

For a tutorial introduction to Phase, see the *Phase Quick Start Guide*. For installation instructions, see the *Installation Guide*.

1.1 Phase Workflows

Phase consists of the following four workflows:

- Building a pharmacophore model (and an optional QSAR models) from a set of ligands
- Building a pharmacophore hypothesis from a single ligand (and editing it)
- Preparing a 3D database that includes pharmacophore information
- Searching the database for matches to a pharmacophore hypothesis

Each of these workflows is supported by a Maestro panel. The first workflow, building a pharmacophore model, involves the following steps:

- Preparing the ligands, including 2D-3D structure conversion and the generation of ligand conformations. This step is described in detail in Chapter 4.
- Defining and identifying the pharmacophore sites in the ligands. This step is described in detail in Chapter 5.
- Creating hypotheses by finding common pharmacophores. This step is described in detail in Chapter 6.
- Scoring the hypotheses, and adding any excluded volumes to the hypotheses. This step is described in detail in Chapter 7.
- Building and examining 3D QSAR models. This step is described in detail in Chapter 8.

Building a pharmacophore hypothesis from one or more ligands manually is an alternative to building a pharmacophore model from a set of ligands by the automated process described above. Details of this task can be found in Chapter 9.

Preparing the 3D database involves one or more of the following tasks, which are described in Chapter 10:

- Preparing the molecules, including 2D-3D conversion
- Generating conformations for each molecule
- Defining and identifying the pharmacophore sites
- Creating subsets

Searching the 3D database for matches to a hypothesis includes various filtering and scoring mechanisms. This workflow is described in Chapter 11.

You can also run these three workflows from the command line, as described in Chapter 12 and Chapter 13. Chapter 14 describes searching a file for matches, rather than a 3D database.

1.2 Citing Phase in Publications

The use of this program should be acknowledged in publications as:

Phase, version 2.0, Schrödinger, LLC, New York, NY, 2005.

Introduction to Maestro

Maestro is the graphical user interface for all of Schrödinger's products: CombiGlide™, Epik™, Glide™, Impact™, Jaguar™, Liaison™, LigPrep™, MacroModel®, Phase™, Prime™, QikProp™, QSite™, SiteMap™, and Strike™. It contains tools for building, displaying, and manipulating chemical structures; for organizing, loading, and storing these structures and associated data; and for setting up, monitoring, and visualizing the results of calculations on these structures. This chapter provides a brief introduction to Maestro and some of its capabilities. For more information on any of the topics in this chapter, see the *Maestro User Manual*.

2.1 General Interface Behavior

Most Maestro panels are amodal: more than one panel can be open at a time, and a panel need not be closed for an action to be carried out. Each Maestro panel has a Close button so you can hide the panel from view.

Maestro supports the mouse functions common to many graphical user interfaces. The left button is used for choosing menu items, clicking buttons, and selecting objects by clicking or dragging. This button is also used for resizing and moving panels. The right button displays a shortcut menu. Other common mouse functions are supported, such as using the mouse in combination with the SHIFT or CTRL keys to select a range of items and select or deselect a single item without affecting other items.

In addition, the mouse buttons are used for special functions described later in this chapter. These functions assume that you have a three-button mouse. If you have a two-button mouse, ensure that it is configured for three-button mouse simulation (the middle mouse button is simulated by pressing or holding down both buttons simultaneously).

2.2 Starting Maestro

Before starting Maestro, you must first set the `SCHRODINGER` environment variable to point to the installation directory. To set this variable, enter the following command at a shell prompt:

csh/tcsh: `setenv SCHRODINGER` *installation-directory*

bash/ksh: `export SCHRODINGER=`*installation-directory*

You might also need to set the DISPLAY environment variable, if it is not set automatically when you log in. To determine if you need to set this variable, enter the command:

```
echo $DISPLAY
```

If the response is a blank line, set the variable by entering the following command:

csh/tcsh: `setenv DISPLAY` *display-machine-name*`:0.0`

bash/ksh: `export DISPLAY=`*display-machine-name*`:0.0`

After you set the SCHRODINGER and DISPLAY environment variables, you can start Maestro using the command:

`$SCHRODINGER/maestro` *options*

If you add the $SCHRODINGER directory to your path, you only need to enter the command maestro. Options for this command are given in Section 2.1 of the *Maestro User Manual*.

The directory from which you started Maestro is Maestro's current working directory, and all data files are written to and read from this directory unless otherwise specified (see Section 2.8 on page 25). You can change directories by entering the following command in the command input area (see page 6) of the main window:

`cd` *directory-name*

where *directory-name* is either a full path or a relative path.

2.3 The Maestro Main Window

The Maestro main window is shown in Figure 2.1 on page 5. The main window components are listed below.

The following components are always visible:

- **Title bar**—displays the Maestro version, the project name (if there is one) and the current working directory.
- **Auto-Help**—automatically displays context-sensitive help.
- **Menu bar**—provides access to panels.
- **Workspace**—displays molecular structures and other 3D graphical objects.

The following components can be displayed or hidden by choosing the component from the Display menu. Your choice of which main window components are displayed is persistent between Maestro sessions.

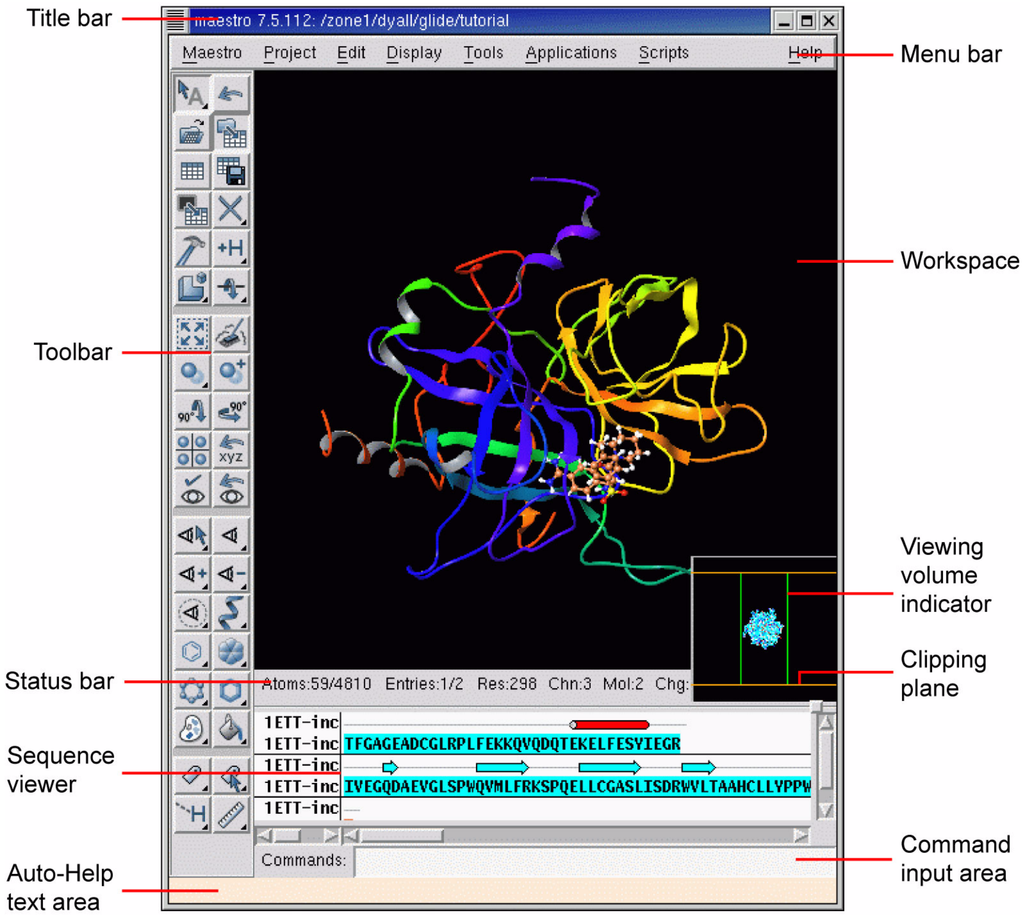


Figure 2.1. The Maestro main window.

- **Toolbar**—contains buttons for many common tasks and provides tools for displaying and manipulating structures, as well as organizing the Workspace.
- **Status bar**—displays information about a particular atom, or about structures in the Workspace, depending on where the pointer pauses (see Section 2.5 of the *Maestro User Manual* for details):
  - **Atom**—displays the chain, residue number, element, PDB atom name, formal charge, and title or entry name (this last field is set by choosing Preferences from the Maestro menu and selecting the Feedback folder).
  - **Workspace**—displays the number of atoms, entries, residues, chains, and molecules in the Workspace.

- **Clipping planes window**—displays a small, top view of the Workspace and shows the clipping planes and viewing volume indicators.
- **Sequence viewer**—shows the sequences for proteins displayed in the Workspace. See Section 2.6 of the *Maestro User Manual* for details.
- **Command input area**—provides a place to enter Maestro commands.

When a distinction between components in the main window and those in other panels is needed, the term *main* is applied to the main window components (e.g., main toolbar).

You can expand the Workspace to occupy the full screen, by pressing CTRL+=. All other components and panels are hidden. To return to the previous display, press CTRL+= again.

2.3.1 The Menu Bar

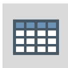
The menus on the main menu bar provide access to panels, allow you to execute commands, and control the appearance of the Workspace. The main menus are as follows:

- Maestro—save or print images in the Workspace, execute system commands, save or load a panel layout, set preferences, set up Maestro command aliases, and quit Maestro.
- Project—open and close projects, import and export structures, make a snapshot, and annotate a project. These actions can also be performed from the Project Table panel. For more information, see Section 2.4 on page 11.
- Edit—undo actions, build and modify structures, define command scripts and macros, and find atoms in the Workspace.
- Display—control the display of the contents of the Workspace, arrange panels, and display or hide main window components.
- Tools—group atoms; measure, align, and superimpose structures; and view and visualize data.
- Applications—set up, submit, and monitor jobs for Schrödinger's computational programs. Some products have a submenu from which you can choose the task to be performed.
- Scripts—manage and install Python scripts that come with the distribution and scripts that you create yourself. (See Chapter 13 of the *Maestro User Manual* for details.)
- Help—open the Help panel, the PDF documentation index, or information panels; run a demonstration; and display or hide Balloon Help (tooltips).

2.3.2 The Toolbar

The main toolbar contains three kinds of buttons for performing common tasks:

Action—Perform a simple task, like clearing the Workspace.



Display—Open or close a panel or open a dialog box, such as the Project Table panel.

Menu—Display a *button menu*. These buttons have a triangle in the lower right corner.

There are four types of items on button menus, and all four types can be on the same menu (see Figure 2.2):

- **Action**—Perform an action immediately.
- **Display**—Open a panel or dialog box.
- **Object types for selection**—Choose Atoms, Bonds, Residues, Chains, Molecules, or Entries, then click on an atom in the Workspace to perform the action on all the atoms in that structural unit.

  The object type is marked on the menu with a red diamond and the button is indented to indicate the action to be performed.
- **Other setting**—Set a state, choose an attribute, or choose a parameter and click on atoms in the Workspace to display or change that parameter.

The toolbar buttons are described below. Some descriptions refer to features not described in this chapter. See the *Maestro User Manual* for a fuller description of these features.

Figure 2.2. The Workspace selection ***button menu and the*** Adjust distances, angles or dihedrals ***button menu.***

Workspace selection
– Choose an object type for selecting
– Open the Atom Selection dialog box

Undo/Redo
Undo or redo the last action. Performs the same function as the Undo item on the Edit menu, and changes to an arrow pointing in the opposite direction when an Undo has been performed, indicating that its next action is Redo.

Open a project
Open the Open Project dialog box.

Import structures
Open the Import panel.

Open/Close Project Table
Open the Project Table panel or close it if it is open.

Save as
Open the Save Project As dialog box, to save the project with a new name.

Create entry from Workspace
Open a dialog box in which you can create an entry in the current project using the contents of the Workspace.

Delete
– Choose an object type for deletion
– Delete hydrogens and waters
– Open the Atom Selection dialog box
– Delete other items associated with the structures in the Workspace
– Click to select atoms to delete
– Double-click to delete all atoms

Open/Close Build panel
Open the Build panel or close it if it is open.

Add hydrogens
– Choose an object type for applying a hydrogen treatment
– Open the Atom Selection dialog box
– Click to select atoms to treat
– Double-click to apply to all atoms

Local transformation
– Choose an object type for transforming
– Click to select atoms to transform
– Open the Advanced Transformations panel

Adjust distances, angles or dihedrals
– Choose a parameter for adjusting
– Delete adjustments

Fit to screen
Scale the displayed structure to fit into the Workspace and reset the center of rotation.

Clear Workspace
Clear all atoms from the Workspace.

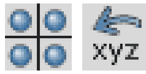
Set fog display state
Choose a fog state. Automatic means fog is on when there are more than 40 atoms in the Workspace, otherwise it is off.

Enhance depth cues
Optimize fogging and other depth cues based on what is in the Workspace.

Rotate around X axis by 90 degrees
Rotate the Workspace contents around the X axis by 90 degrees.

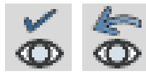
Rotate around Y axis by 90 degrees
Rotate the Workspace contents around the Y axis by 90 degrees.

Tile entries
Arrange entries in a rectangular grid in the Workspace.



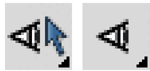
Reset Workspace
Reset the rotation, translation, and zoom of the Workspace to the default state.

Save view
Save the current view of the Workspace: orientation, location, and zoom.



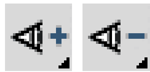
Restore view
Restore the last saved view of the Workspace: orientation, location, and zoom.

Display only selected atoms
– Choose an object type for displaying
– Click to select atoms to display
– Double-click to display all atoms



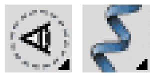
Display only
– Choose a predefined atom category
– Open the Atom Selection dialog box

Also display
– Choose a predefined atom category
– Open the Atom Selection dialog box



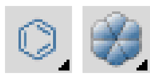
Undisplay
– Choose a predefined atom category
– Open the Atom Selection dialog box

Display residues within N angstroms of currently displayed atoms
– Choose a radius
– Open a dialog box to set a value



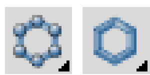
Show, hide, or color ribbons
– Choose to show or hide ribbons
– Choose a color scheme for coloring ribbons

Draw bonds in wire
– Choose an object type for drawing bonds in wire representation
– Open the Atom Selection dialog box
– Click to select atoms for representation
– Double-click to apply to all atoms



Draw atoms in CPK
– Choose an object type for drawing bonds in CPK representation
– Open the Atom Selection dialog box
– Click to select atoms for representation
– Double-click to apply to all atoms

Draw atoms in Ball & Stick
– Choose an object type for drawing bonds in Ball & Stick representation
– Open the Atom Selection dialog box
– Click to select atoms for representation
– Double-click to apply to all atoms

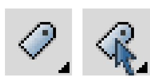


Draw bonds in tube
– Choose an object type for drawing bonds in tube representation
– Open the Atom Selection dialog box
– Click to select atoms for representation
– Double-click to apply to all atoms

Color all atoms by scheme
Choose a predefined color scheme.

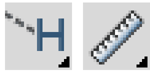
Color residue by constant color
– Choose a color for applying to residues
– Click to select residues to color
– Double-click to color all atoms

Label atoms
– Choose a predefined label type
– Delete labels



Label picked atoms
– Choose an object type for labeling atoms
– Open the Atom Selection dialog box
– Open the Atom Labels panel at the Composition folder
– Delete labels
– Click to select atoms to label
– Double-click to label all atoms

Display H-bonds
– Choose bond type:
intra—displays H-bonds within the selected molecule
inter—displays H-bonds between the selected molecule and all other atoms.
– Delete H-bonds
– Click to select molecule



Measure distances, angles or dihedrals
– Choose a parameter for displaying measurements
– Delete measurements
– Click to select atoms for measurement

2.3.3 Mouse Functions in the Workspace

The left mouse button is used for selecting objects. You can either click on a single atom or bond, or you can drag to select multiple objects. The right mouse button opens shortcut menus, which are described in Section 2.7 of the *Maestro User Manual*.

The middle and right mouse buttons can be used on their own and in combination with the SHIFT and CTRL keys to perform common operations, such as rotating, translating, centering, adjusting, and zooming.

Table 2.1. Mapping of Workspace operations to mouse actions.

Mouse Button	Keyboard	Motion	Action
Left		click, drag	Select
Left	SHIFT	click, drag	Toggle the selection
Middle		drag	Rotate about X and Y axes Adjust bond, angle, or dihedral
Middle	SHIFT	drag vertically	Rotate about X axis
Middle	SHIFT	drag horizontally	Rotate about Y axis
Middle	CTRL	drag horizontally	Rotate about Z axis
Middle	SHIFT + CTRL	drag horizontally	Zoom
Right		click	Spot-center on selection
Right		click and hold	Display shortcut menu
Right		drag	Translate in the X-Y plane
Right	SHIFT	drag vertically	Translate along the X axis
Right	SHIFT	drag horizontally	Translate along the Y axis
Right	CTRL	drag horizontally	Translate along the Z axis
Middle & Right		drag horizontally	Zoom

2.3.4 Shortcut Key Combinations

Some frequently used operations have been assigned shortcut key combinations. The shortcuts available in the main window are described in Table 2.2.

Table 2.2. Shortcut keys in the Maestro main window.

Keys	Action	Equivalent Menu Choices
CTRL+B	Open Build panel	Edit > Build
CTRL+C	Create entry	Project > Create Entry From Workspace
CTRL+E	Open Command Script Editor panel	Edit > Command Script Editor
CTRL+F	Open Find Atoms panel	Edit > Find
CTRL+H	Open Help panel	Help > Help
CTRL+I	Open Import panel	Project > Import Structures
CTRL+M	Open Measurements panel	Tools > Measurements
CTRL+N	Create new project	Project > New
CTRL+O	Open project	Project > Open
CTRL+P	Print	Maestro > Print
CTRL+Q	Quit	Maestro > Quit
CTRL+S	Open Sets panel	Tools > Sets
CTRL+T	Open Project Table panel	Project > Show Table
CTRL+W	Close project	Project > Close
CTRL+Z	Undo/Redo last command	Edit > Undo/Redo
CTRL+=	Enter and exit full screen mode (Workspace occupies full screen)	None

2.4 Maestro Projects

All the work you do in Maestro is done within a *project*. A project consists of a set of *entries*, each of which contains one or more chemical structures and their associated data. In any Maestro session, there can be only one Maestro project open. If you do not specify a project when you start Maestro, a *scratch* project is created. You can work in a scratch project without saving it, but you must save it in order to use it in future sessions. When you save or close a project, all the view transformations (rotation, translation, and zoom) are saved with it. When you close a project, a new scratch project is automatically created.

Likewise, if there is no entry displayed in the Workspace, Maestro creates a *scratch* entry. Structures that you build in the Workspace constitute a scratch entry until you save the structures as project entries. The scratch entry is not saved with the project unless you explicitly add it to the project. However, you can use a scratch entry as input for some calculations.

To add a scratch entry to a project, do one of the following:

- Click the Create entry from Workspace button:

- Choose Create Entry from Workspace from the Project menu.
- Press CTRL+C.

In the dialog box, enter a name and a title for the entry. The entry name is used internally to identify the entry and can be modified by Maestro. The title can be set or changed by the user, but is not otherwise modified by Maestro.

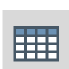
Once an entry has been incorporated into the project, its structures and their data are represented by a row in the Project Table. Each row contains the row number, an icon indicating whether the entry is displayed in the Workspace (the In column), the entry title, a button to open the Surfaces panel if the entry has surfaces, the entry name, and any entry properties. The row number is not a property of the entry.

Entries can be collected into groups, and the members of the group can be displayed or hidden. Most additions of multiple entries to the Project Table are done as entry groups.

You can use entries as input for all of the computational programs—Glide, Impact, Jaguar, Liaison, LigPrep, MacroModel, Phase, Prime, QikProp, QSite, and Strike. You can select entries as input for the ePlayer, which displays the selected structures in sequence. You can also duplicate, combine, rename, and sort entries; create properties; import structures as entries; and export structures and properties from entries in various formats.

To open the Project Table panel, do one of the following:

- Click the Open/Close Project Table button on the toolbar



- Choose Show Table from the Project menu
- Press CTRL+T.

The Project Table panel contains a menu bar, a toolbar, and the table itself.

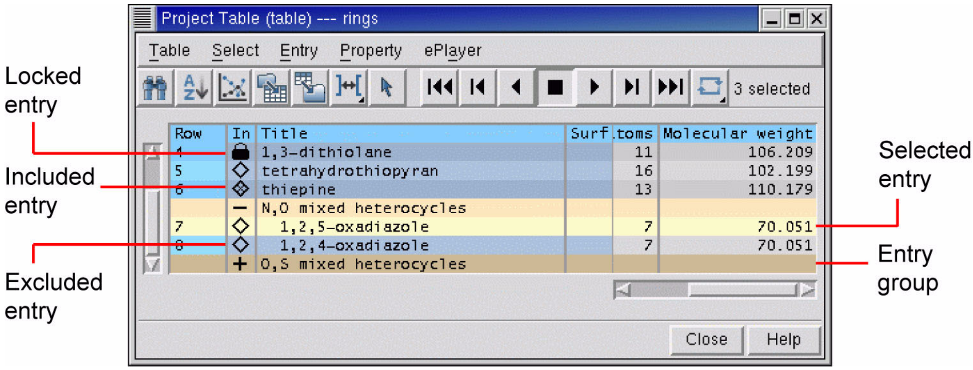
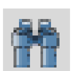


Figure 2.3. The Project Table ***panel.***

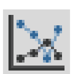
2.4.1 The Project Table Toolbar

The Project Table toolbar contains two groups of buttons and a status display. The first set of buttons opens various panels that allow you to perform functions on the entries in the Project Table. The second set of buttons controls the ePlayer, which “plays through” the selected structures: each structure is displayed in the Workspace in sequence, at a given time interval. See Section 2.3.2 on page 7 for a description of the types of toolbar buttons. The buttons are described below.



Find
Open the Find panel for locating alphanumeric text in any column of the Project Table, except for the row number.

Sort
Open the Sort panel for sorting entries by up to three properties.



Plot
Open the Plot panel for plotting entry properties.

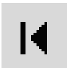
Import Structure
Open the Import panel for importing structures into the project.

Export Structure
Open the Export panel for exporting structures to a file.

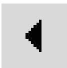
Columns
Choose an option for adjusting the column widths.

Select only
Open the Entry Selection dialog box for selecting entries based on criteria for entry properties.

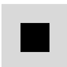
Go to start
Display the first selected structure.



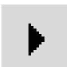
Previous
Display the previous structure in the list of selected structures.



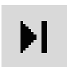
Play backward
Display the selected structures in sequence, moving toward the first.



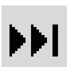
Stop
Stop the ePlayer.



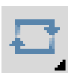
Play forward
Display the selected structures in sequence, moving toward the last.



Next
Display the next structure in the list of selected structures.



Go to end
Display the last selected structure.



Loop
Choose an option for repeating the display of the structures. Single Direction displays structures in a single direction, then repeats. Oscillate reverses direction each time the beginning or end of the list is reached.

The status display, to the right of the toolbar buttons, shows the number of selected entries. When you pause the cursor over the status display, the Balloon Help shows the total number of entries, the number shown in the table, the number selected, and the number included in the Workspace.

2.4.2 The Project Table Menus

- Table—find text, sort entries, plot properties, import and export structures, and configure the Project Table.
- Select—select all entries, none, invert your selection, or select classes of entries using the Entry Selection dialog box and the Filter panel.
- Entry—include or exclude entries from the Workspace, display or hide entries in the Project Table, and perform various operations on the selected entries.
- Property—display and manipulate entry properties in the Project Table.
- ePlayer—view entries in succession, stop, reverse, and set the ePlayer options.

2.4.3 Selecting Entries

Many operations in Maestro are performed on the entries selected in the Project Table. The Project Table functions much like any other table: select rows by clicking, shift-clicking, and control-clicking. However, because clicking in an editable cell of a selected row enters edit mode, you should click in the Row column to select entries. See Section 2.4.5 on page 16 for more information on mouse actions in the Project Table. There are shortcuts for selecting classes of entries on the Select menu.

In addition to selecting entries manually, you can select entries that meet a combination of conditions on their properties. Such combinations of conditions are called *filters*. Filters are Entry Selection Language (ESL) expressions and are evaluated at the time they are applied. For example, if you want to set up a Glide job that uses ligands with a low molecular weight (say, less than 300) and that has certain QikProp properties, you can set up a filter and use it to select entries for the job. If you save the filter, you can use it again on a different set of ligands that meet the same selection criteria.

To create a filter:

1. Do one of the following:
   - Choose Only, Add, or Deselect from the Select menu.
   - Click the Entry selection button on the toolbar.

2. In the Properties folder, select a property from the property list, then select a condition.
3. Combine this selection with the current filter by clicking Add, Subtract, or Intersect. These buttons perform the Boolean operations OR, AND NOT, and AND on the corresponding ESL expressions.
4. To save the filter for future use click Create Filter, enter a name, and click OK.
5. Click OK to apply the filter immediately.

2.4.4 Including Entries in the Workspace

In addition to selecting entries, you can also use the Project Table to control which entries are displayed in the Workspace. An entry that is displayed in the Workspace is *included* in the Workspace; likewise, an entry that is not displayed is *excluded*. Included entries are marked by an X in the diamond in the In column; excluded entries are marked by an empty diamond. Entry inclusion is completely independent of entry selection.

To include or exclude entries, click, shift-click, or control-click in the In column of the entries, or select entries and choose Include or Exclude from the Entry menu. Inclusion with the mouse works just like selection: when you include an entry by clicking, all other entries are excluded.

It is sometimes useful to keep one entry in the Workspace and include others one by one: for example, a receptor and a set of ligands. You can fix the receptor in the Workspace by selecting it in the Project Table and choosing Fix from the Entry menu or by pressing CTRL+F. A padlock icon replaces the diamond in the In column to denote a *fixed* entry. To remove a fixed entry from the Workspace, you must exclude it explicitly (CTRL+X). It is not affected by the inclusion or exclusion of other entries. Fixing an entry affects only its inclusion; you can still rotate, translate, or modify the structure.

2.4.5 Mouse Functions in the Project Table

The Project Table supports the standard use of shift-click and control-click to select objects. This behavior applies to the selection of entries and the inclusion of entries in the Workspace. You can also drag to resize rows and columns and to move rows.

You can drag a set of non-contiguous entries to reposition them in the Project Table. When you release the mouse button, the entries are placed after the first unselected entry that precedes the entry on which the cursor is resting. For example, if you select entries 2, 4, and 6, and release the mouse button on entry 3, these three entries are placed after entry 1, because entry 1 is the first unselected entry that precedes entry 3. To move entries to the top of the table, drag them above the top of the table; to move entries to the end of the table, drag them below the end of the table.

A summary of mouse functions in the Project Table is provided in Table 2.3.

Table 2.3. Mouse operations in the Project Table.

Task	Mouse Operation
Change a Boolean property value	Click repeatedly in a cell to cycle through the possible values (On, Off, Clear)
Display the Entry menu for an entry	Right-click anywhere in the entry. If the entry is not selected, it becomes the selected entry. If the entry is selected, the action is applied to all selected entries.
Display a version of the Property menu for a property	Right-click in the column header
Edit the text or the value in a table cell	Click in the cell and edit the text or value
Include an entry in the Workspace, exclude all others	Click the In column of the entry

Table 2.3. Mouse operations in the Project Table. (Continued)

Task	Mouse Operation
Move selected entries	Drag the entries
Paste text into a table cell	Middle-click
Resize rows or columns	Drag the boundary with the middle mouse button
Select an entry, deselect all others	For an unselected entry, click anywhere in the row except the In column; for a selected entry, click the row number.
Select or include multiple entries	Click the first entry then shift-click the last entry
Toggle the selection or inclusion state	Control-click the entry or the In column

2.4.6 Project Table Shortcut Keys

Some frequently used project operations have been assigned shortcut key combinations. The shortcuts, their functions, and their menu equivalents are listed in Table 2.4.

Table 2.4. Shortcut keys in the Project Table.

Keys	Action	Equivalent Menu Choices
CTRL+A	Select all entries	Select > All
CTRL+F	Fix entry in Workspace	Entry > Fix
CTRL+I	Open Import panel	Table > Import Structures
CTRL+N	Include only selected entries	Entry > Include Only
CTRL+U	Deselect all entries	Select > None
CTRL+X	Exclude selected entries	Entry > Exclude
CTRL+Z	Undo/Redo last command	Edit > Undo/Redo in main window

2.5 Building a Structure

After you start Maestro, the first task is usually to create or import a structure. You can open existing Maestro projects or import structures from other sources to obtain a structure, or you can build your own. To open the Build panel, do one of the following:

- Click the Open/Close Build panel button in the toolbar:

- Choose Build from the Edit menu.
- Press CTRL+B.

The Build panel allows you to create structures by drawing or placing atoms or fragments in the Workspace and connecting them into a larger structure, to adjust atom positions and bond orders, and to change atom properties. This panel contains a toolbar and three folders.

2.5.1 Placing and Connecting Fragments

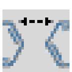
The Build panel provides several tools for creating structures in the Workspace. You can place and connect fragments, or you can draw a structure freehand.

To place a fragment in the Workspace:

1. Select Place.
2. Choose a fragment library from the Fragments menu.
3. Click a fragment.
4. Click in the Workspace where you want the fragment to be placed.

To connect fragments in the Workspace, do one of the following:

- Place another fragment and connect them using the Connect & Fuse panel, which you open from the Edit menu on the main menu bar or with the Display Connect & Fuse panel on the Build toolbar.



- Replace one or more atoms in the existing fragment with another fragment by selecting a fragment and clicking in the Workspace on the main atom to be replaced.
- Grow another fragment by selecting Grow in the Build panel and clicking the fragment you want to add in the Fragments folder.

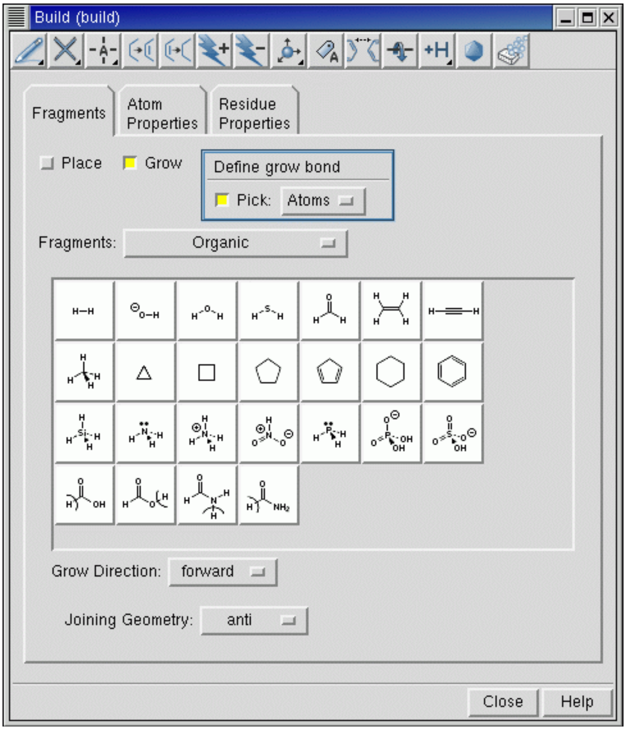
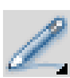


Figure 2.4. The Build ***panel.***

Grow mode uses predefined rules to connect a fragment to the *grow bond*. The grow bond is marked by a green arrow. The new fragment replaces the atom at the head of the arrow on the grow bond and all atoms attached to it. To change the grow bond, choose Bonds from the Pick option menu in the Build panel and click on the desired grow bond in the Workspace. The arrow points to the atom nearest to where you clicked.

To draw a structure freehand:

1. Choose an element from the Draw button menu on the Build panel toolbar:



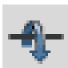
2. Click in the Workspace to place an atom of that element.
3. Click again to place another atom and connect it to the previous atom.
4. Continue this process until you have drawn the structure.
5. Click the active atom again to finish drawing.

2.5.2 Adjusting Properties

In the Atom Properties folder, you can change the properties of the atoms in the Workspace. For each item on the Property option menu—Element, Atom Type (MacroModel), Partial Charge, PDB Atom Name, Grow Name, and Atom Name—there is a set of tools you can use to change the atom properties. For example, the Element tools consist of a periodic table from which you can choose an element and select an atom to change it to an atom of the selected element.

Similarly, the Residue Properties folder provides tools for changing the properties of residues: the Residue Number, the Residue Name, and the Chain Name.

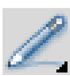
To adjust bond lengths, bond angles, dihedral angles, and chiralities during or after building a structure, use the Adjust distances, angles or dihedrals button on the main toolbar:



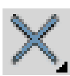
You can also open the Adjust panel from this button menu, from the Display Adjust panel button on the Build panel toolbar (which has the same appearance as the above button) or from the Edit menu in the main window.

2.5.3 The Build Panel Toolbar

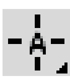
The toolbar of the Build panel provides quick access to tools for drawing and modifying structures and labeling atoms. See Section 2.3.2 on page 7 for a description of the types of toolbar buttons. The toolbar buttons and their use are described below.



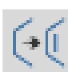
Free-hand drawing
Choose an element for drawing structures freehand in the Workspace (default C). Each click in the Workspace places an atom and connects it to the previous atom.



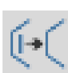
Delete
Choose an object for deleting. Same as the Delete button on the main toolbar, see page 8.



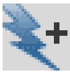
Set element
Choose an element for changing atoms in the Workspace (default C). Click an atom to change it to the selected element.



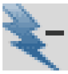
Increment bond order
Select a bond to increase its bond order by one, to a maximum of 3.



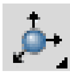
Decrement bond order
Select a bond to decrease its bond order by one, to a minimum of 0.



Increment formal charge
Select an atom to increase its formal charge by one.

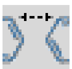


Decrement formal charge
Select an atom to decrease its formal charge by one.

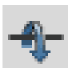


Move
Choose a direction for moving atoms, then click the atom to be moved. Moves in the XY plane are made by clicking the new location. Moves in the Z direction are made in 0.5 Å increments.

Label
Apply heteroatom labels as you build a structure. The label consists of the element name and formal charge, and is applied to atoms other than C and H.



Display Connect & Fuse panel
Open the Connect & Fuse panel so you can connect structures (create bonds between structures) or fuse structures (replace atoms of one structure with those of another).



Display Adjust panel
Open the Adjust panel so you can change bond lengths, bond angles, dihedral angles, or atom chiralities.

Add hydrogens
Choose an atom type for applying the current hydrogen treatment. Same as the Add hydrogens button on the main toolbar, see page 8.

Geometry Symmetrizer
Open the Geometry Symmetrizer panel for symmetrizing the geometry of the structure in the Workspace.

Geometry Cleanup
Clean up the geometry of the structure in the Workspace.

2.6 Selecting Atoms

Maestro has a powerful set of tools for selecting atoms in a structure: toolbar buttons, picking tools in panels, and the Atom Selection dialog box. These tools allow you to select atoms in two ways:

- Select atoms first and apply an action to them
- Choose an action first and then select atoms for that action

2.6.1 Toolbar Buttons

The small triangle in the lower right corner of a toolbar button indicates that the button contains a menu. Many of these buttons allow you to choose an object type for selecting: choose Atoms, Bonds, Residues, Chains, Molecules, or Entries, then click on an atom in the Workspace to perform the action on all the atoms in that structural unit.

For example, to select atoms with the Workspace selection toolbar button:

1. Choose Residues from the Workspace selection button menu:

   The button changes to:

2. Click on an atom in a residue in the Workspace to select all the atoms in that residue.

2.6.2 Picking Tools

The picking tools are embedded in each panel in which you need to select atoms to apply an operation. The picking tools in a panel can include one or more of the following:

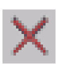
- Pick option menu—Allows you to choose an object type. Depending on the operation to be performed, you can choose Atoms, Bonds, Residues, Chains, Molecules, or Entries, then click on an atom in the Workspace to perform the action on all the atoms in that structural unit.

  The Pick option menu varies from panel to panel, because not all object types are appropriate for a given operation. For example, some panels have only Atoms and Bonds in the Pick option menu.

- All button—Performs the action on all atoms in the Workspace.
- Selection button—Performs the action on any atoms already selected in the Workspace.
- Previous button—Performs the action on the most recent atom selection defined in the Atom Selection dialog box.
- Select button—Opens the Atom Selection dialog box.
- ASL text box—Allows you to type in an ASL expression for selecting atoms.

  ASL stands for Atom Specification Language, and is described in detail in the *Maestro Command Reference Manual*.

- Clear button—Clears the current selection

- Show markers option—Marks the selected atoms in the Workspace.

For example, to label atoms with the Label Atoms panel:

1. Choose Atom Labels from the Display menu.
2. In the Composition folder, select Element and Atom Number.
3. In the picking tools section at the top of the panel, you could do one of the following:
   - Click Selection to apply labels to the atoms already selected in the Workspace (from the previous example).
   - Choose Residues from the Pick option menu and click on an atom in a different residue to label all the atoms in that residue.

2.6.3 The Atom Selection Dialog Box

If you wish to select atoms based on more complex criteria, you can use the Atom Selection dialog box. To open this dialog box, choose Select from a button menu or click the Select button in a panel. See Section 5.3 of the *Maestro User Manual* for detailed instructions on how to use the Atom Selection dialog box.

2.7 Scripting in Maestro

Although you can perform nearly all Maestro-supported operations through menus and panels, you can also perform operations using Maestro commands, or compilations of these commands, called *scripts*. Scripts can be used to automate lengthy procedures or repetitive tasks and can be created in several ways. These are summarized below.

2.7.1 Python Scripts

Python is a full-featured scripting language that has been embedded in Maestro to extend its scripting facilities. The Python capabilities within Maestro include access to Maestro functionality for dealing with chemical structures, projects, and Maestro files.

The two main Python commands used in Maestro are:

- `pythonrun`—executes a Python module. (You can also use the alias `pyrun`.) The syntax is:

  `pythonrun` *module*.*function*

- `pythonimport`—rereads a Python file so that the next time you use the `pythonrun` command, it uses the updated version of the module. (You can also use the alias `pyimp`.)

From the Maestro Scripts menu you can install, manage, and run Python scripts. For more information on the Scripts menu, see Section 13.1 of the *Maestro User Manual*.

For more information on using Python with Maestro, see *Scripting with Python*.

2.7.2 Command Scripts

All Maestro commands are logged and displayed in the Command Script Editor panel. This means you can create a command script by performing the operations with the GUI controls, copying the logged commands from the Command History list into the Script text area of the panel, then saving the list of copied commands as a script.

To run an existing command script:

1. Open the Command Script Editor panel from the Edit menu in the main window.
2. Click Open Local and navigate to the directory containing the desired script.
3. Select a script in the Files list and click Open.

   The script is loaded into the Script window of the Command Script Editor panel.
4. Click Run Script.

Command scripts cannot be used for Prime operations.

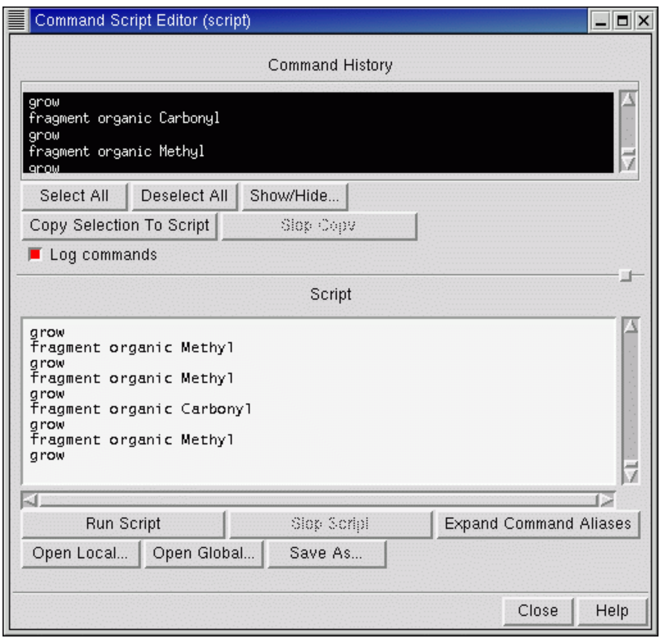


Figure 2.5. The Command Script Editor ***panel.***

2.7.3 Macros

There are two kinds of macros you can create: named macros and macros assigned to function keys F1 through F12.

To create and run a named macro:

1. Open the Macros panel from the Edit menu in the main window.
2. Click New, enter a name for the macro, and click OK.
3. In the Definition text box, type the commands for the macro.
4. Click Update to update the macro definition.
5. To run the macro, enter the following in the command input area in the main window:

   `macrorun` *macro-name*

   If the command input area is not visible, choose Command Input Area from the Display menu.

To create and run a function key macro:

1. Open the Function Key Macros panel from the Edit menu in the main window.
2. From the Macro Key option, select a function key (F1 through F12) to which to assign the macro.
3. In the text box, type the commands for the macro.
4. Click Run to test the macro or click Save to save it.
5. To run the macro from the main window, press the assigned function key.

For more information on macros, see Section 13.5 of the *Maestro User Manual*.

2.8 Specifying a Maestro Working Directory

When you use Maestro to launch Phase jobs, Maestro writes job output to the directory specified in the Directory folder of the Preferences panel. By default, this directory (the file I/O directory) is the directory from which you started Maestro.

To change the Maestro working directory:

1. Open the Preferences panel from the Maestro menu.
2. Click the Directory tab.
3. Select the directory you want to use for reading and writing files.

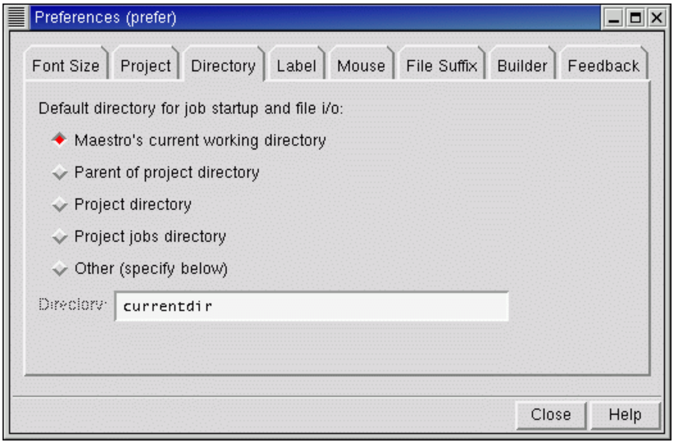


Figure 2.6. The Directory ***folder of the*** Preferences ***panel.***

You can also set other preferences in the Preferences panel. See Section 12.2 of the *Maestro User Manual* for details.

2.9 Undoing an Operation

To undo a single operation, click the Undo button in the toolbar, choose Undo from the Edit menu, or press CTRL+Z. The word Undo in the menu is followed by text that describes the operation to undo. Not all operations can be undone: for example, global rotations and translations are not undoable operations. For such operations you can use the Save view and Restore view buttons in the toolbar, which save and restore a molecular orientation.

2.10 Running and Monitoring Jobs

Maestro has panels for each product for preparing and submitting jobs. To use these panels, choose the appropriate product and task from the Applications menu and its submenus. Set the appropriate options in the panel, then click Start to open the Start dialog box and set options for running the job. For a complete description of the Start dialog box associated with your computational program, see your product's User Manual. When you have finished setting the options, click Start to launch the job and open the Monitor panel.

The Monitor panel is the control panel for monitoring the progress of jobs and for pausing, resuming, or killing jobs. All jobs that belong to you can be displayed in the Monitor panel, whether or not they were started from Maestro. Subjobs are indented under their parent in the job list. The text pane shows output information from the monitored job, such as the contents of the log file. The Monitor panel opens automatically when you start a job. If it is not open, you can open it by choosing Monitor from the Applications menu in the Maestro main window.

While jobs are running, the Detach, Pause, Resume, Stop, Kill, and Update buttons are active. When there are no jobs currently running, only the Monitor and Delete buttons are active. These buttons act on the selected job. By default, only jobs started from the current project are shown. To show other jobs, deselect Show jobs from current project only.

When a monitored job ends, the results are incorporated into the project according to the settings used to launch the job. If a job that is not currently being monitored ends, you can select it in the Monitor panel and click Monitor to incorporate the results. Monitored jobs are incorporated only if they are part of the current project. You can monitor jobs that are not part of the current project, but their results are not incorporated. To add their results to a project, you must open the project and import the results.

Further information on job control, including configuring your site, monitoring jobs, running jobs, and job incorporation, can be found in the *Job Control Guide* and the *Installation Guide*.

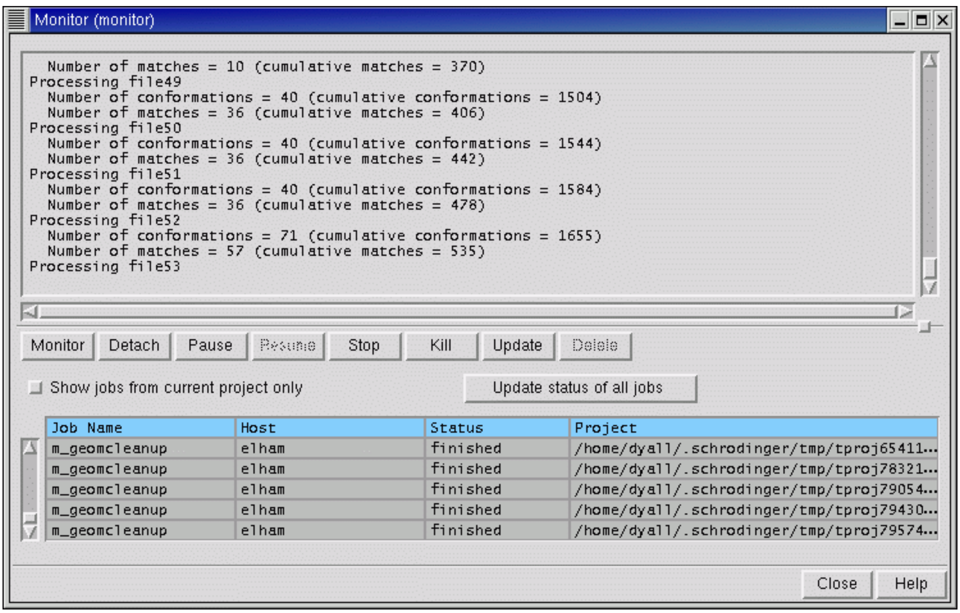


Figure 2.7. The** Monitor **panel.

2.11 Getting Help

Maestro comes with automatic, context-sensitive help (Auto-Help), Balloon Help (tooltips), an online help facility, and a user manual. To get help, follow the steps below:

- Check the Auto-Help text box at the bottom of the main window. If help is available for the task you are performing, it is automatically displayed there. It describes what actions are needed to perform the task.
- If your question concerns a GUI element, such as a button or option, there may be Balloon Help for the item. Pause the cursor over the element. If the Balloon Help does not appear, check that Show Balloon Help is selected in the Help menu of the main window. If there is Balloon Help for the element, it appears within a few seconds.
- If you do not find the help you need using either of the steps above, click the Help button in the lower right corner of the appropriate panel. The Help panel is displayed with a relevant help topic.
- For help with a concept or action not associated with a panel, open the Help panel from the Help menu or press CTRL+H.

If you do not find the information you need in the Maestro help system, check the following sources:

- The *Maestro User Manual*
- The Frequently Asked Questions page on the Schrödinger Support Center.

You can also contact Schrödinger by e-mail or phone for help:

- E-mail: help@schrodinger.com
- Phone: (503) 299-1150

2.12 Ending a Maestro Session

To end a Maestro session, choose Quit from the Maestro menu. To save a log file with a record of all operations performed in the current session, click Quit, save log file in the Quit panel. This information can be useful to Schrödinger support staff when responding to any problem you report.

Running Phase from Maestro

Phase consists of the following workflows, each of which is supported by a Maestro panel:

- Building a pharmacophore model and an optional QSAR model
- Preparing a 3D database that includes pharmacophore information
- Building or editing pharmacophore hypotheses
- Searching the database for matches to a pharmacophore hypothesis

An overview of each of these workflows is given in the sections below, along with an overview of the supporting Maestro panel. The stages are described in detail in the following chapters.

The Maestro interface for the first workflow is wizard-like, and takes you through each step of the process. The interface is more flexible than a wizard, however, because it allows you to exit at any step and resume the process later at the same step-or a different step, provided that you have the data to support that step. Default options are provided that should give good results, but you can also choose from a range of other options to suit your purposes. The interface for the remaining three workflows are single panels.

To open any of the panels, Develop Pharmacophore Model, Manage 3D Database, Edit Hypotheses, or Find Matches to Hypotheses, choose the appropriate item from the Phase submenu of the Applications menu in the main window.

All Phase jobs can be started from Maestro. Many of these jobs generate a large amount of data, and most of them can be distributed across multiple processors. When you click a Start button or an action button that starts a job, a panel is displayed that allows you to set job control information.

Phase can also be run from the command line. For information on command-line use, see Chapter 12, Chapter 13, and Chapter 14.

3.1 Developing a Pharmacophore Model

Developing a pharmacophore model from a set of active molecules is the main means in Phase of generating pharmacophore hypotheses, which are subsequently used in database searching.

In Phase, there are five steps in the process of developing a pharmacophore model: preparing the ligands, creating pharmacophore sites from a set of features, finding common pharmacophores, scoring the hypotheses, and building a QSAR model.

The process of developing a pharmacophore model is called a *run*. The data for each run is stored as a separate entity, which you can open from the File menu. When a new run is created, links are made to common data, to avoid duplication. The run is stored as part of a Maestro project.

3.1.1 General Panel Layout

The Develop Pharmacophore Model panel is wizard-like in design, with five steps. Each step occupies the center of the panel, and consists of a title with a brief description of the step at the top, a set of controls and tables for results in the center, and a Back and a Next button at the bottom. The features of each step are described in detail in the online help.

In addition to the step features, the panel contains a menu bar, a toolbar, and an octagon icon at the top; and a step guide (the Guide) at the bottom above the Close and Help buttons. These features are described below.

The File Menu

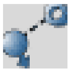
The File menu allows you to work with the runs that are available in the project.

New	Create a new run
Open	Open an existing run from the submenu. If there are more than 4 runs, choose More to open a dialog box and select a run.
Save As	Save the current run with a new name
Rename	Rename the current run
Delete Run	Delete the current run

The Display Menu and the Toolbar

The Display menu provides options for viewing hypotheses and related attributes in the Workspace. These options are also available as toolbar buttons, and are described below. The items are only available in Step 4 (Score Hypotheses) and Step 5 (Build QSAR Model).

	Hypothesis	Displays the selected hypothesis as a spatial arrangement of feature symbols. For a description of these symbols, see Table 5.1 on page 51.
	Excluded Volumes	Displays excluded volumes for the selected hypothesis.
	QSAR Model	Displays the selected QSAR model for the hypothesis. Only available in the Build QSAR Model step.

 Distances		Display markers for the intersite distances in the selected hypothesis.
 Angles		Display markers for the angles in the selected hypothesis.

The Step Menu

The Step menu contains an item to display or hide the Guide, and items for each of the steps. The current step is marked with a red diamond. If the Guide is displayed, it is marked with a red square. The items for the steps that are not available are dimmed. You can go to any available step by choosing the corresponding menu item.

The Octagon Button

When a job has been launched and is running, the gray octagon at the upper right of the panel turns green and spins. To monitor the job using the Monitor panel, click the octagon. For more information about monitoring jobs, see Chapter 2 or the *Job Control Guide*.

The Guide

The Guide displays the steps in the model as a set of buttons linked by lines. The buttons for the steps that are not available are dimmed. The current step is highlighted with a white background. You can go to any available step by clicking its button in the Guide. The Guide can be displayed or hidden from the Step menu. If you go back to an earlier step and make changes, you are prompted to save the existing data and create a new run with the changed data.

To navigate the steps, you can click the Back and Next buttons, click the desired step in the Guide, or choose the desired step from the Step menu.

3.1.2 The Prepare Ligands Step

In this step, you add to the run the molecules that you want to use as the basis for the pharmacophore model and any other molecules that you want to use to build or test the QSAR model, and you select both active and inactive molecules for the set that is used for the pharmacophore model. If you include activity data with the ligands when you add them, you can select the active and the inactive sets of ligands either using cutoffs or manually.

To develop a pharmacophore model, you should ensure that you have all-atom 3D structures, and generate different conformations for each structure. If the structures need to be converted from 2D to 3D or otherwise need cleaning up, you can do so in this step. You can also convert the structures to the most probable ionization (protonation) state at a given pH, and generate different chiralities for the structures in this step. The molecules that you add are automatically grouped into conformer sets. You can choose to group stereoisomers in the same set or different sets. If you add only one conformer for a given molecule, you can generate the rest in this step.

Once you have added the molecules, cleaned up the structures, generated conformers, and selected the set of ligands, you can proceed to the next step.

3.1.3 The Create Sites Step

In this step, you use a set of chemical structure patterns to identify pharmacophore features in each ligand. Once a feature has been mapped to a specific location in a conformation, it is referred to as a *pharmacophore site*. The number of occurrences of each feature in each ligand is tabulated, and you can display the locations of the pharmacophore sites in the Workspace.

While the built-in set of features is adequate for many purposes, you might want to add new patterns to the built-in features, ignore patterns in the built-in features, or add custom features. You can add functional groups, defined as SMARTS patterns, to the definition of a feature. You can also designate functional groups that should be excluded from consideration as part of a feature, and you can choose to ignore functional groups.

3.1.4 The Find Common Pharmacophores Step

In this step, you perform a search for common pharmacophores among the set of high-affinity (active) ligands that you chose in the first step. The search spans one or more families of pharmacophores, known as *variants*. You can choose the number of site points in the pharmacophore, filter out variants that have too many or too few of a particular kind of feature, and select a set of variants from the filtered list. You can also set a lower limit on the number of ligands that must match a pharmacophore before it can be considered to be a hypothesis.

The search proceeds by enumerating all pharmacophores of a given variant and partitioning them into successively smaller high-dimensional boxes according to their intersite distances. Each n-point pharmacophore contains $n(n–1)/2$ unique intersite distances, so each box contains $n(n–1)/2$ dimensions. Pharmacophores that are clustered into the same box are considered to be equivalent and therefore common to the ligands from which they arise. The size of the box defines the tolerance on each intersite distance, and therefore how similar common pharmacophores must be. You can set parameters to control the minimum box size and you can exclude pharmacophores for which any intersite distance is below a certain threshold.

Boxes that contain pharmacophores from the minimum required number of ligands are said to *survive* the partitioning process. Each surviving box contains a set of common pharmacophores, one of which is ultimately singled out as a hypothesis.

Once all desired variants have been processed, you can continue to the scoring step.

3.1.5 The Score Hypotheses Step

In this step you apply a scoring function that identifies the best candidate hypothesis from each surviving box, and provides an overall ranking of all the hypotheses. You can finish at this point, or select hypotheses for the generation of QSAR models and continue to the next step, or select hypotheses and proceed to find matches to the hypotheses. You can also add to the hypothesis volumes that should not be occupied by atoms in any active molecule, known as *excluded volumes*.

The scoring algorithm includes contributions from the alignment of site points and vectors, volume overlap, selectivity, number of ligands matched, relative conformational energy, and activity. You can adjust these in the survival score, and you can create a custom score. You can also penalize hypotheses by scoring inactives and subtracting a multiple of this score from the survival score. The scores for each hypothesis are displayed in a table. You can select a hypothesis and view scores for the ligands that match the hypothesis, and the energy of the ligand relative to the lowest conformation.

If you have both active and inactive molecules that match a hypothesis, you can use their structures to define excluded volumes. Any region of space that is occupied by part of an inactive molecule and is not occupied by the active molecules is a good candidate for an excluded volume. The excluded volumes are used to filter out molecules in the database search that are likely to be inactive.

When you have selected one or more hypotheses, you can proceed to the next step.

3.1.6 The Build QSAR Model Step

In this step, you build QSAR models for the selected hypotheses using the activity data for molecules that match at least three points in the hypothesis. You can use molecules with any level of activity, including those that may be inactive due to steric clashes with the target receptor. The QSAR model partitions space into a grid of uniformly sized cubes, and characterizes each molecule by a set of binary-valued independent variables that encode the occupancy of these cubes by six atom classes or a set of pharmacophore feature types. Partial least-squares regression is applied to these variables to build a series of models with successively greater numbers of factors.

You can visualize the QSAR model in the Workspace, and analyze it by atom or feature class and ligand. This can be used to identify ligand features that contribute positively or negatively to the predicted activity.

When you have developed QSAR models, you can continue to the database search and use the QSAR models to predict activities for matches, or return to the previous step to select another set of hypotheses for QSAR model development.

3.2 Building or Editing Hypotheses

As an alternative to building a pharmacophore model, you can build pharmacophore hypotheses directly from known active molecules, in the Edit Hypotheses panel. In this task, Phase identifies the possible pharmacophore sites in the molecule you select, based on a set of pharmacophore feature definitions. You then select the features that you want to include in the hypothesis. No jobs are run in this workflow.

3.3 Preparing a 3D Database for Searching

The Manage 3D Database panel provides tools for preparing a structure database that can be searched for matches to a pharmacophore hypothesis. The database must contain all-atom 3D structures that are reasonable representations of the experimental structures. Preparing a database involves adding structures, cleaning the structures if necessary, generating conformers if necessary or desired, creating pharmacophore sites from selected features, and creating subsets of molecules for database searching. The last step is optional.

Databases are not connected to Maestro projects.

The Manage 3D Database panel is a single panel, with a menu bar and an octagon button at the top. When a job has been launched and is running, the gray octagon at the upper right of the panel turns green and spins. To monitor the job using the Monitor panel, click the octagon. For more information about monitoring jobs, see Chapter 2 or the *Job Control Guide*.

3.4 Finding Matches to a Hypothesis

The Find Matches to Hypothesis panel is a single panel, with four sections. In the top two sections, you specify the database to search and the hypothesis to use in the search. In the bottom two sections, you set parameters for the search and for the subsequent display of hits.

The search is performed in two steps: finding matches to the hypothesis, and fetching hits. The second step can be repeated with different processing options without repeating the first step. The processing options include adjusting the fitness score, by which hits are sorted, applying numerical cutoffs on the number of hits, applying excluded volumes to filter hits, and calculating activities using the QSAR model, if one was generated for the hypothesis.

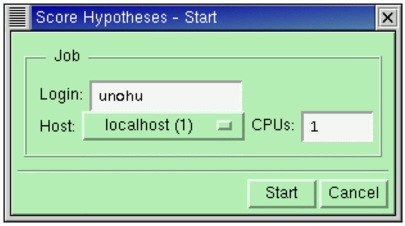


Figure 3.1. The* Start *dialog box.

3.5 Running Jobs

When you click an action button that is associated with a job or a Start button, the Start dialog box opens. In this dialog box, you can select the host on which you want to run the job, set the user name on that host, if it differs from the user name on the host on which you are running Maestro, and enter the number of processors to use for the job. The maximum number of processors available on the selected host is displayed in parentheses after the host name.

Phase jobs run under the Schrödinger job control facility. This facility allows you to monitor the progress of jobs within Maestro, both local and remote. It also provides the list of hosts in the Start panel from which you make a selection when you start a job.

The list of hosts is read from the `schrodinger.hosts` file, which is installed in the `$SCHRODINGER` directory. At installation time, this file should be set up to define the hosts on which Schrödinger software will be run. Instructions for setting up this file are given in the *Installation Guide*. You can copy this file to your home directory to customize it.

The time-consuming parts of Phase can be distributed across multiple processors. You can set up multiprocessor hosts in the `schrodinger.hosts` file, either as hosts on which you run jobs directly or as batch queues, with a specified number of processors. If you run a database search on a multiprocessor host, such as a cluster, the following requirements must be met:

- The database must be located in a directory that is uniformly accessible to all nodes of the cluster on which jobs will be run.
- If the file system where the database is stored is only accessible to the cluster, you must run Maestro on the manager node of the cluster to launch jobs.
- In the `$SCHRODINGER/schrodinger.hosts` file, each parallel queue that is used for database jobs should have a `tmpdir` entry with a path that is accessible to all nodes. For details of setting up these entries, see the *Installation Guide*.

Preparing Ligands for Pharmacophore Model Development

The first step in developing a pharmacophore model is to select the molecules that you want to use and to prepare them for use. This step is performed in the Prepare Ligands step of the Develop Pharmacophore Model panel.

The molecules you select should at a minimum include highly active molecules that you want to use as the basis of the pharmacophore model. You can also include inactive or moderately active molecules, which can be used to test pharmacophore hypotheses for specificity, for building and testing a QSAR model, and for the purpose of defining excluded volumes. When you add molecules, you can select an associated activity property to use for activity scoring and for the dependent variable in the QSAR model. This property can also be used to define the active and inactive molecules to use as the basis for the pharmacophore model.

Developing a pharmacophore model requires all-atom 3D structures that are realistic representations of the experimental molecular structure. Most ligands are flexible, so it is important to consider a range of conformations in order to increase the chances of finding something close to the bound structure.

Under some circumstances it can be important to generate variations on the input structures, such as varying the chirality or choosing the most probable protonation state. Varying the chirality of atoms in the molecules can be important if the chiralities are not known. The process of identifying common pharmacophores can then sample the different stereoisomers and locate the one that matches best. Also, if the input structure is not in the most common form at physiological pH values, the identification of common pharmacophores might give incorrect results, because the active form is protonated or deprotonated.

In the Prepare Ligands step, you add the molecules with their activity values to the Phase run. If you have 2D structures or united-atom 3D structures, you can convert them to all-atom 3D structures and locate the minimum energy structure using molecular mechanics. In the process, you can vary the chirality of atoms in the structures and assign the protonation state. Once you have the structures and any variations, you can generate the low-energy conformers for each structure. The conformers are automatically grouped into sets for each molecule. If you already have all-atom, 3D structures with their conformations, you must still pass them through the ligand preparation steps, so that Phase can verify that there are no unusable structures and that the geometry of the structures is sufficiently accurate.

The tasks involved in this step and how to accomplish them are described in detail below.

4.1 Adding Ligands to a Run

Each "ligand" in Phase is actually a set of conformations of a ligand structure. When you add ligands, they are automatically grouped into conformer sets. If you add ligands from a previous run, the sets are preserved with all their associated data. If you do not have conformations for the ligands you add, you can generate them in this step.

Stereoisomers can be considered as "the same ligand" or "different ligands." You can group stereoisomers into different sets or into the same set by selecting or deselecting the Separate Stereoisomers option. By default, stereoisomers are treated as separate ligands, because their activities are often very different.

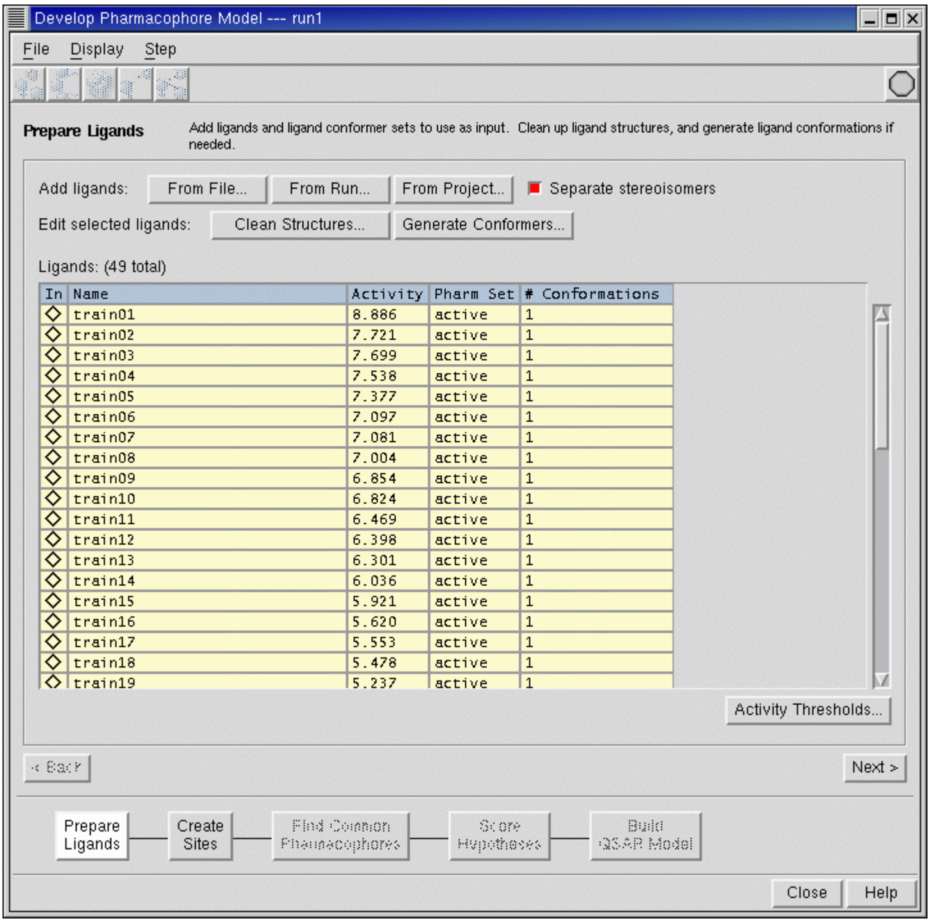


Figure 4.1. ***The*** Prepare Ligands ***step.***

You can add ligands to a Phase run from a file, from a previous run, or from a Maestro project. To add ligands, click one of the Add Ligands buttons. Each button opens a dialog box in which you can read or copy the ligands. Activity data can be added with the ligands. Once you have added the ligands, they are displayed in the Ligands table. If the ligands do not have activity data, you can add the data by editing the table cells.

If you want to delete ligands, select them in the table, then right-click in the table and choose Delete from the contextual menu.

Adding Ligands From a File

You can read ligands directly from a file into the Phase run, without importing them into Maestro. To do so, click From File. The Add From File dialog box is displayed. This dialog box is similar to the Import panel. In it you can select a contiguous set of structures from a file and read structures from multiple files. The entire range of file formats available from the Import panel is available in this dialog box. Properties are read with the structures. When you click Add, the Choose Activity Property dialog box opens. This dialog box contains a list of properties, from which you can choose a single property for the activity of the ligands, and convert the activity to a logarithmic scale if necessary. The activity must be a positive quantity that increases with increasing activity.

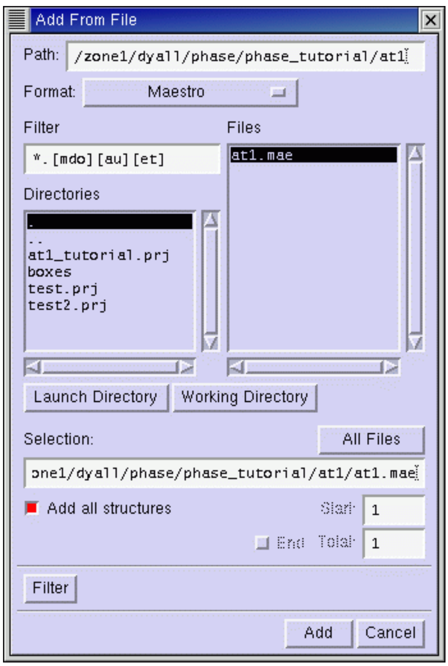


Figure 4.2. The Add From File ***dialog box.***

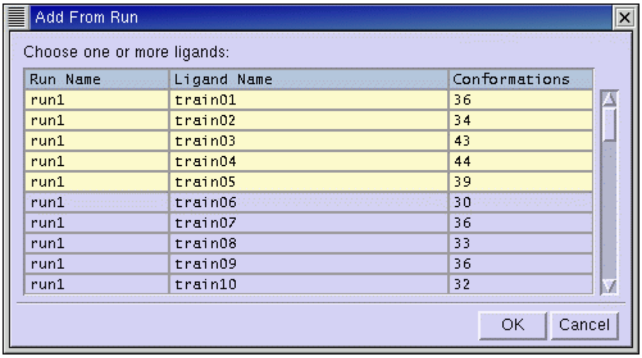


Figure 4.3. The Add From Run ***dialog box.***

Adding Ligands From Another Phase Run

To copy ligands from another Phase run in the current project, click From Run. The Add From Run dialog box is displayed. The dialog box contains a list of all ligands available from all other runs in the current project, with the run name, the ligand name, and the number of conformers. You can choose multiple ligands to add to the current run. The activity values and the membership of the active set are extracted and added with the ligand. If a ligand was used in more than one run, the list of ligands will contain duplicates. If you select duplicates, only one is added, with the activity data from the first run chosen.

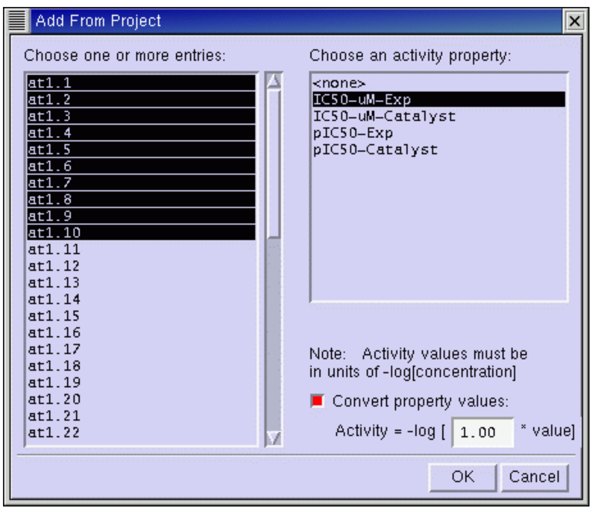


Figure 4.4. The Add From Project ***dialog box.***

Adding Ligands From the Project

If you already have ligands in the Maestro project that you want to use, you can copy them from the project into the Phase run. To do so, click From Project. The Add From Project dialog box is displayed. This dialog box contains two lists: a list of entries, and a list of properties.

You can choose multiple entries to be added to the Ligands table. By default, the entries that are selected in the Project Table are selected in the entry list in the dialog box. The ligands are copied into the run, so that any changes made by Phase have no effect on the original ligands in the Project Table.

You can choose a single property for the activity of the ligands. The ligand activity must be a property that has units of $-\log_{10}[x]$, where x is the ligand. If the property is in units of concentration [x], you can convert it to a logarithmic scale in this dialog box. The converted values are copied to the Ligands table.

4.2 Cleaning Up Ligand Structures

If the ligand structures are two-dimensional, lack hydrogen atoms, or include counter ions or solvent molecules, you must clean them up before proceeding. If the structures do not have the desired chirality or ionization (protonation) state, or if you want structures with different chirality, you can use the Clean Structures facility to generate them. Clean Structures is an interface to LigPrep with a range of options that is most appropriate for Phase. For more detailed information about the process, see the *LigPrep User Manual*.

In the cleanup process, the following actions are performed as necessary or as requested:

- Convert structures from 2D to 3D
- Add hydrogen atoms to ensure that the structure is an all-atom structure
- Remove counter ions and water molecules
- Add or remove protons to produce the most probable ionization state at the target pH
- Generate stereoisomers
- Remove noncompliant structures
- Perform an energy minimization

The cleanup process is applied to the ligands that are selected in the Ligands table. You can therefore perform the cleanup with different options for different sets of molecules by making different selections. To clean up the selected structures, click Clean Structures. The Clean Structures dialog box is displayed. In this dialog box, you can set options for generation of stereoisomers and ionization states, then click Start to run the job to perform the cleanup. When you click Start, a dialog box is displayed, in which you can choose the host to run the job. You can distribute this job over multiple hosts.

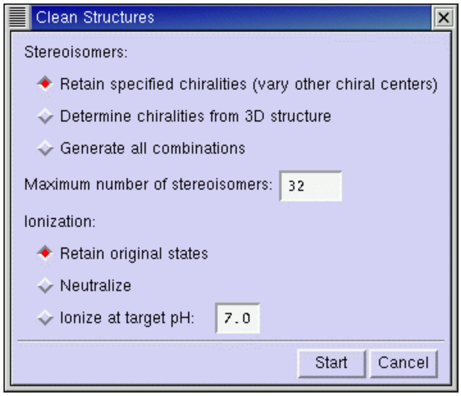


Figure 4.5. The Clean Structures ***dialog box.***

Generating Stereoisomers

There are three options for generating stereoisomers, described below. For each option, any unspecified chiralities are varied, up to the number given in the Maximum number of stereoisomers text box. When you vary the stereochemistry, the process starts at the configuration with all chiral atoms to be varied set to R, and systematically varies the configuration. If you select fewer stereoisomers than the maximum, there is a chance that you might not generate the most important stereoisomers.

Retain specified chiralities (vary other chiral centers)

If the ligand has chirality information, this information is retained and used to ensure that the chiral atoms all have the correct chiralities. Chirality information includes parities and bond directions from SD files and the chirality property from Maestro files. If the configuration or chirality of one or more chiral centers is not specified, the chiralities for these centers is varied.

Determine chiralities from 3D structures

This option discards any information from the input file and determines the chirality from the 3D geometry. These chiralities are held fixed. For centers whose chirality is indeterminate, the two possible chiralities are generated.

Generate all combinations

This option discards chirality information and generates all possible configurations that result from the combination of chiralities on each chiral center.

Generating Ionization States

In the Ionization section you can choose from three options for generating the appropriate ionization state:

Retain original states

This option bypasses the generation of ionization states. If the ligands all have the correct ionization state for acidic and basic groups, choose this option.

Neutralize

This option converts all acidic and basic groups into their neutral form. For example, zwitterion groups are converted from a carboxylate and an ammonium to a carboxylic acid and an amine.

Ionize at target pH

This option generates the most probable ionization state at the given target pH, for which the default value is 7.

4.3 Generating Conformers

Once you have a set of cleaned-up ligands, you can run a conformational search to generate a set of conformers for each ligand. If you already have the conformations you need, you can skip this step.

To set up parameters for the conformational search, click Generate Conformers. The Generate Conformers dialog box is displayed. The dialog box has options for the search mode and solvation treatment, and allows you to limit the number of conformations generated, either to a specific number or by energy, which is evaluated in aqueous solution with a continuum solvation model. After setting options, click Start to run the conformational search job. A dialog box is displayed, in which you can choose the host to run the job. You can distribute this job over multiple processors. When the job finishes, the Ligands table displays the number of conformers generated for each ligand.

Some options have a greater impact than others on the outcome of pharmacophore model development. Options with the greatest impact include the maximum number of conformations, maximum relative energy difference, minimum atomic deviation and number of post-minimization iterations. The default settings—rapid search, distance-dependent dielectric solvation model, and no post-minimization iterations—are likely to be adequate for many purposes. However, for consistency, you should use the same options in the pharmacophore model development as you use in the database search.

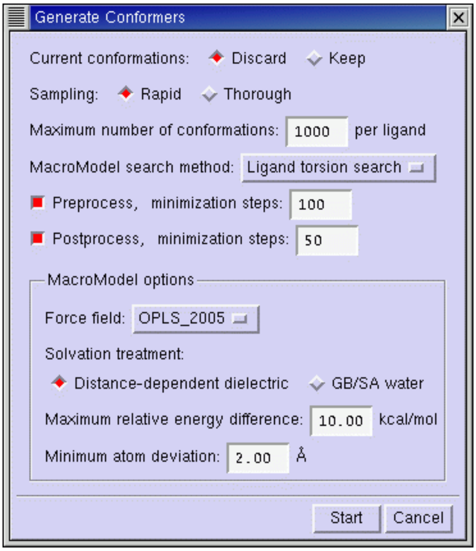


Figure 4.6. The Generate Conformers ***dialog box.***

The options for controlling the conformational search are described below.

4.3.1 Output Options

Current conformations

When you generate conformations, you can discard the existing conformer set or you can keep it. If you keep the existing set, the new conformations are appended to the set. The set might therefore contain redundant conformers.

Maximum number of conformations

This value limits the number of conformers returned from the generation process. If the number of conformers generated is higher than this value, a sample of all the conformers generated is returned.

4.3.2 Search Method and Sampling Options

Conformer generation can be performed with one of two search methods, Ligand torsion search or Mixed MCMM/LMOD. For each method, sampling of conformational space can be done in Rapid or Thorough mode. Experience to date suggests that the final pharmacophore model is not usually significantly improved by a thorough search.

During the search, hydrogen-bonding interactions are suppressed, because conformations in which the ligand bonds to the receptor are needed in the model, not just conformations with internal hydrogen bonding.

Ligand torsion search

In the ligand torsional search, the molecule is divided into a core and a periphery. The peripheral groups have only one rotatable bond between the terminal groups and the rest of the molecule. All the nonperipheral rotatable bonds are assigned to the core. The conformational search generates all core configurations and then varies the peripheral configurations, either one-by-one or in a complete search. The Sampling options have the following meanings:

- Rapid—All the core conformations are generated, and the conformations of the peripheral (rotamer) groups are sampled one by one.
- Thorough—A complete set of conformations is generated for both the core and the peripheral groups.

Mixed MCMM/LMOD

The alternative search method is a combined Monte-Carlo Multiple Minimum/Low Mode (MCMM/LMOD) search, and is more accurate than the ligand torsional sampling method, but as a consequence takes longer. The difference between Rapid and Thorough sampling is in the number of steps taken per rotatable bond, which is much larger for thorough sampling. For more information on this method, see Chapter 10 of the *MacroModel User Manual*.

4.3.3 Preprocessing and Postprocessing Options

Both the Preprocess and the Postprocess options perform a MacroModel minimization, using the options in the MacroModel options section. Preprocessing is done on the input structure; postprocessing is done on the set of conformers generated by the conformational search. The same set of minimization options is used in both cases. These two options each have a text box in which you can set the number of iterations to perform in the minimization of the structures. If you set the number to zero, only the energy is calculated: no minimization is performed. The minimization and the energy calculation are done with MacroModel using the selected force field.

In addition to minimization, postprocessing filters out redundant conformers and conformers that are high in energy. If you want to filter the generated conformers but not minimize them, set the number of minimization steps for postprocessing to zero. The MacroModel energy is calculated for the conformers and is used to eliminate high-energy conformers. Minimization of the conformers generated by a mixed MCMM/LMOD search is recommended.

If you do not minimize the energy, the generation of conformers runs much faster. Many of the conformers are rejected because their energy is too high, so the number of conformers is usually smaller than if you do the energy minimization.

4.3.4 MacroModel Options

In this section, you can select the force field and solvation treatment, and set thresholds to limit the number of conformations generated and determine when two conformers are considered to be identical.

Force Field

The default force field is OPLS_2005, but you can also select MMFFs. For details on these force fields, see Section 3.1 of the *MacroModel User Manual*.

Solvation treatment

Two continuum solvation treatments for water are provided.

- Distance-dependent dielectric
- GB/SA water

The distance-dependent dielectric model is somewhat faster than the GB/SA model, and usually produces similar results.

Maximum relative energy difference

This value sets an energy threshold relative to the lowest-energy conformer. Conformers that are higher in energy than this threshold are discarded. The energy is evaluated with MacroModel using the selected force field.

Minimum atom deviation

All distances between pairs of corresponding heavy atoms must be below this threshold for two conformers to be considered identical. This threshold is only applied after the energy difference threshold, and only if the two conformers are within 1 kcal/mol of each other.

In addition to this threshold, a threshold of 60° is used for torsion angle differences for polar hydrogens. This threshold cannot be changed.

4.4 The Ligands Table

The Ligands table lists the ligands that you added, grouped into conformer sets. You can select table rows in the usual way with click, shift-click and control-click. You can sort the columns by clicking the column header, and you can resize the columns by dragging the column boundary. If you right-click in the table, a menu is displayed. From this menu you can select all ligands, invert the selection, delete the selected ligands, or export them to the Project Table or to a file.

The table columns are described in Table 4.1.

Table 4.1. Description of the Ligand table columns.

Column	Description
In	Inclusion status of the ligand. The diamond has a cross in it if the ligand is included in the Workspace, and is empty if the ligand is excluded. The molecule that is displayed is the first conformer of the set. To view other conformers, you must export them to the Project Table (right-click menu). This column functions like the In column of the Project Table: click in the diamond to include a ligand and exclude all others, control-click to include or exclude a ligand without affecting the inclusion of the others, and shift-click to include a range of ligands. The included ligands are added as a scratch entry to the Workspace. Inclusion and exclusion of ligands has no effect on the entries in the Project Table.
Name	The name of the ligand. The default name is taken from the Title property of the ligand, if you added it from a project or from a file in which the title is defined. Otherwise a name is created for the ligand. You can edit the name by clicking in the cell, changing the text, then pressing ENTER. The name does not have to be unique.
Activity	Contains the value of the activity you selected when you added the ligands. If you did not select an activity, the table cells are empty. You can edit the activity by clicking in the cell, changing the text, then pressing ENTER.
Pharm Set	Indicates whether a ligand is in the set of active molecules or the set of inactive molecules that will be used to develop the pharmacophore model (the “pharm set”), or is ignored. For these three states the column contains the text active or inactive, or is blank. You can cycle through these states by clicking the table cell. To cycle through the states for all selected rows, control-click any of the selected cells.
# Conformations	The number of conformations stored for the ligand. You will normally want to generate multiple conformers for each ligand, unless, for example, you are developing a pharmacophore model from x-ray structures.

4.5 Defining the Ligand Set for Model Development

There are two ways in which you can define the set of ligands (the "pharm set") that will be used for model development: by setting a threshold, and by manual selection. The ligand set must include some active ligands, and can also include inactive ligands. The ligands marked as active in the Pharm Set column of the Ligands table will be used to develop the model.

To set thresholds for active and inactive ligands, click Activity Thresholds. In the Activity Thresholds dialog box, you can set a threshold for the active ligands and a threshold for the inactive ligands. Ligands with activity greater than or equal to the active threshold are marked as active and included in the pharm set. Ligands with activity less than the inactive threshold are marked as inactive and included in the pharm set. Ligands whose activity lies between the thresholds are excluded from the pharm set.

To add ligands manually to the pharm set, select the ligands (using click, shift-click, or control-click), then control-click the Pharm Set column of the Ligands table. This action changes the status of all selected ligands; a click or a shift-click changes the status of a single ligand.

Note that it is not always necessary to assign every active molecule to the pharm set. If you have groups of highly similar ligands with nearly the same level of activity, you may want to select only one or two ligands from each group. You might also want to reserve some active ligands to test QSAR models.

4.6 Step Summary

To prepare the ligands for pharmacophore model development, follow the steps below.

1. Import the ligands into the Phase run, by clicking From File, From Run, or From Project.
2. Select or deselect Separate stereoisomers, depending on how you want to treat stereoisomers.
3. If you want to build a QSAR model or perform activity scoring, enter activity data for the ligands if it is not already present.
4. Clean up the ligand structures and generate variations on stereochemistry or ionization state by clicking Clean Ligands.
5. Generate sets of conformers for each ligand by clicking Generate Conformers.
6. Define the pharm set, either by setting the activity thresholds (click Activity Threshold), or by selecting ligands in the Ligands table.
7. Click Next to proceed to the next step.

Creating Pharmacophore Sites

The second step in developing a pharmacophore model is to use a set of pharmacophore features to create pharmacophore sites (site points) for all the ligands. This step is performed in the Create Sites step of the Develop Pharmacophore Model panel.

Phase supplies a built-in set of six pharmacophore features:

- Hydrogen bond acceptor (A)
- Hydrogen bond donor (D)
- Hydrophobic group (H)
- Negatively charged group (N)
- Positively charged group (P)
- Aromatic ring (R)

Each pharmacophore feature is defined by a set of chemical structure patterns. All user-defined patterns are specified as SMARTS queries and assigned one of three possible geometries, which define physical characteristics of the site:

- Point—the site is located on a single atom in the SMARTS query.
- Vector—the site is located on a single atom in the SMARTS query, and it will be assigned directionality according to one or more vectors originating from the atom.
- Group—the site is located at the centroid of a group of atoms in the SMARTS query. For aromatic rings, the site is assigned directionality defined by a vector that is normal to the plane of the ring.

Before proceeding, it is important to point out the difference between a *vector feature* and *vector geometry*. "Vector feature" is a more general term that refers to any pharmacophore feature that contains directionality. This includes hydrogen-bond acceptors, hydrogen-bond donors and aromatic rings. "Vector geometry" is more specific, and refers to the particular types of directionality associated with hydrogen-bond acceptors and donors. Thus vector geometry implies vector feature, but vector feature does not necessarily imply vector geometry.

While the built-in feature definitions are adequate for many purposes, you may find it necessary to expand them to include new patterns. For example, the presence of electron-withdrawing groups may cause an otherwise non-acidic hydrogen to be significantly dissociated at pH 7. If the built-in negative ionic definitions do not cover this case, then you may want to supplement the definitions with the appropriate SMARTS pattern.

In some cases, you may feel that a particular built-in definition should not be used, so you can choose to ignore it. Or there may be instances where a built-in definition matches a functionality that you feel does not qualify. In that case you can add a pattern to exclude the functionality in question.

You may also wish to add your own custom features types (X, Y, Z) to account for chemical functionalities that are not covered by the built-in feature types (A, D, H, N, P, R), or to lend special significance to a particular type of pharmacophoric element. If, for example, you know that all actives must contain a piperidine ring, then you could define a custom feature X with a corresponding SMARTS pattern to match piperidine. Or, perhaps you want to force the pharmacophore model to map C=O acceptors only to other C=O acceptors. This could be achieved by creating a custom acceptor feature Y that contains only the SMARTS pattern for C=O.

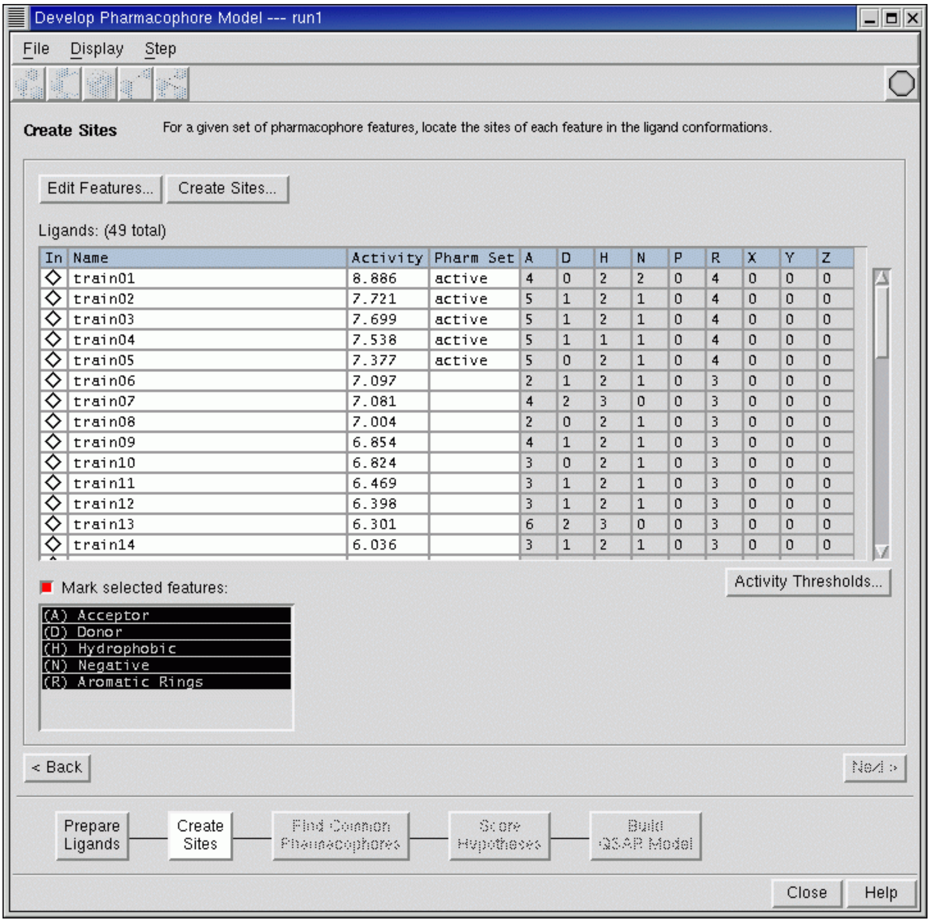


Figure 5.1. The Create Sites ***step.***

The pharmacophore features can be previewed in the Workspace for any ligand. This allows you to verify that the definitions of the features are what you expect, before proceeding to generate site points for the entire set of ligand conformations.

5.1 Viewing Pharmacophore Features

Before you change the definitions of pharmacophore features, or submit the job to create site points using the pharmacophore features, you might want to view the features for each of the ligands. In this way you can check that the features are correctly identified.

Displaying features requires the creation of site points for one conformation of each ligand. This is done automatically when you enter the Create Sites step. If you change the feature definitions, you can create these site points and view the features by clicking Preview. In either case, a job is run locally to create the sites for the first conformer of each ligand. When the job is done (it should be quick), the feature counts are entered in the columns of the Ligands table, and you can display the features in the Workspace.

The first four columns of the Ligands table are the same as in the Prepare Ligands step, and have the same behavior; the selection behavior of the rows is the same, and the right-click menu is the same—see Section 4.4 on page 47 for a description. In place of the Conformations column is a series of columns, one for each pharmacophore feature. These columns are populated with feature counts (the number of times a feature is present in a ligand).

To display a ligand and its pharmacophore features in the Workspace, click the In column of the Ligands table for the ligand, select Mark selected features, and choose the feature types from the list below this option. You can select multiple features from the list. The appearance of the features is described in Table 5.1. To view features for a different ligand, include it in the Workspace using the In column of the Ligands table.

Table 5.1. Visual appearance of pharmacophore features in the Workspace.

Feature	Appearance
Acceptor (A)	Light red sphere centered on the atom with the lone pair, with arrows pointing in the direction of the lone pairs
Donor (D)	Light blue sphere centered on the H atom, with an arrow pointing in the direction of the potential H-bond
Hydrophobic (H)	Green sphere
Negative (N)	Red sphere
Positive (P)	Blue sphere
Aromatic Ring (R)	Orange torus in the plane of the ring
Custom	Colored sphere, with a unique color

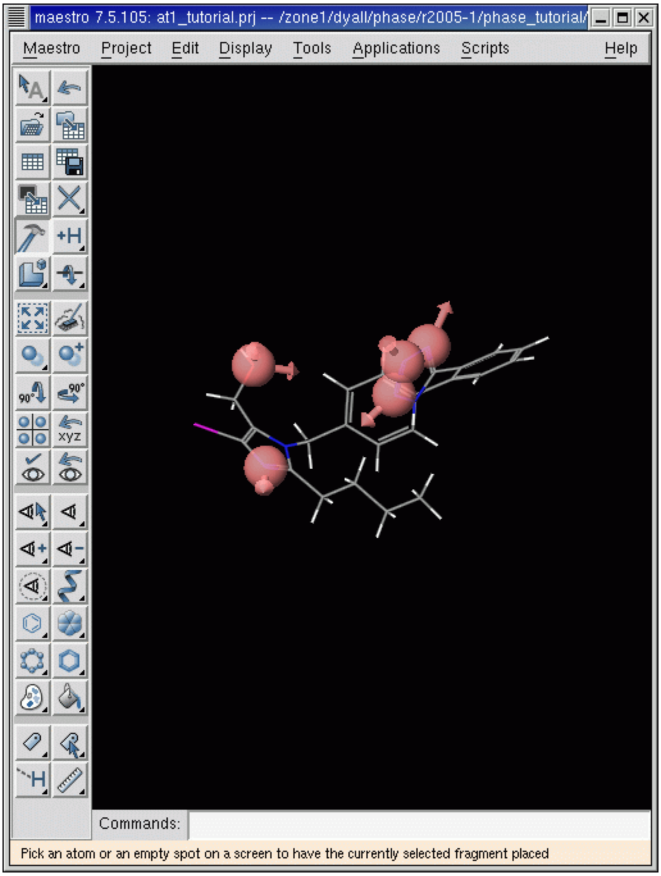


Figure 5.2. Pharmacophore features.

5.2 Editing Pharmacophore Features

If you want to supplement the built-in features, create custom features, or load features from another location, you can do so in the Edit Features dialog box. Features are defined in terms of SMARTS patterns. You can add patterns to both standard features and up to three custom features. You can edit and delete custom patterns, and you can exclude or ignore both standard and custom patterns in a feature.

To open the Edit Features dialog box, click Edit Features.

The Edit Features dialog box, displayed in Figure 5.3, contains a section for loading and storing feature sets, described in the next section, and a section for defining features. The feature definition section has controls for selecting, adding and deleting features, a table listing the SMARTS patterns that define the feature, and controls for adding, deleting, and moving patterns. The Pattern list table lists all the patterns that are used to define the pharmacophore feature. You can only select one row at a time in the table, and the text fields are not editable. The table columns are described in Table 5.2.

5.2.1 Loading and Storing Feature Sets

The built-in feature sets are stored in the Phase product distribution. To reload them, click Reset All. This button also clears the custom sets.

Feature sets are stored with each run. To import a feature set from another run, click Import from Run, and select the desired run in the dialog box that is displayed.

Feature sets can also be stored in a file. As you do not have access to runs from other projects, you must store feature sets that you want to use in other projects in a file. To save a feature set to a file, click Export, and specify the file location in the file chooser that is displayed. To import a feature set from a file, click Import from File, and navigate to the feature file.

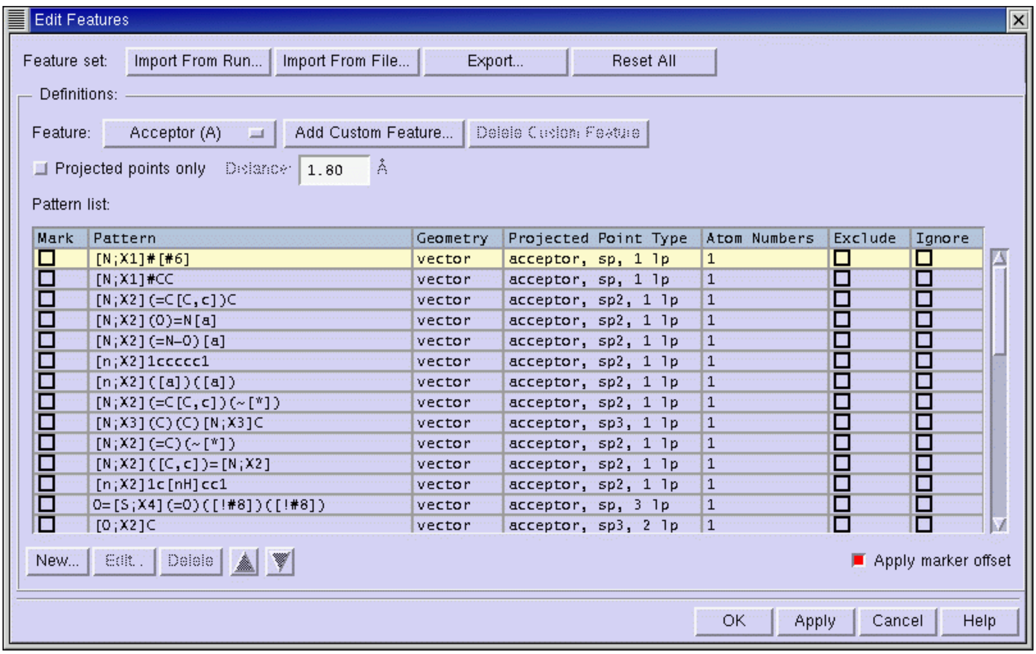


Figure 5.3. The* Edit Features *dialog box.

Table 5.2. Pattern list table columns

Column	Description
Mark	Column of check boxes. Selecting a check box marks the pattern on any ligands that are displayed in the Workspace.
Pattern	Pattern definition. With the exception of default hydrophobic features and aromatic rings, the definitions are all SMARTS strings.
Geometry	Designates physical characteristics of site. Can be point, vector, or group, as described previously.
Projected Point Type	Defines the directionality of vector features. Can be an aromatic ring, a donor, an acceptor with one or more lone pairs, or none (i.e., a nonvector feature).
Atom Numbers	The list of atoms that determine the location of the pharmacophore site, numbered according to the SMARTS string. Point and vector geometries use a single atom, whereas group geometry uses multiple atoms.
Exclude	Column of check boxes. Selecting a check box excludes the atoms in this definition from being mapped by other definitions. This is essentially a NOT operator. Excluded patterns are processed first when searching for features.
Ignore	Column of check boxes. Selecting a check box ignores the pattern when searching for features. Equivalent to deleting the pattern, but keeps the pattern in the table.

5.2.2 Adding and Editing Custom Patterns

If the patterns in a given feature do not cover all the functional groups that you want to include in the feature, you can add extra patterns. To add a new SMARTS pattern to a feature, choose the feature from the Feature option menu, then click the New button below the Pattern list table. The New Pattern dialog box is displayed.

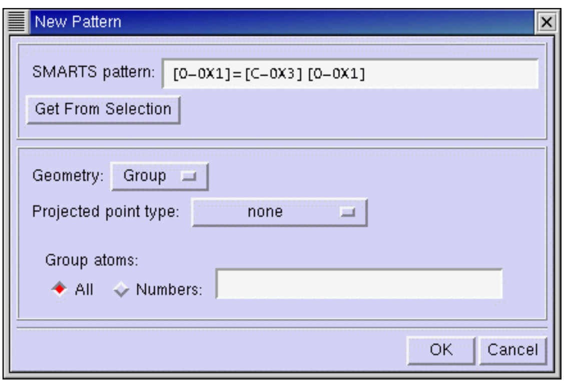


Figure 5.4. The New Pattern ***dialog box.***

In the New Pattern dialog box, you can enter a SMARTS pattern, define the feature geometry and projected point type and the atoms that represent the feature. When you have made these choices, click OK to add the pattern. These choices are described in detail below.

To add a pattern to a feature, you must provide the SMARTS string for the desired arrangement of atoms, and define the corresponding pharmacophore site. The pharmacophore site can be a group, such as an aromatic ring; a single point, such as an atom; or a vector, such as a hydrogen bond acceptor or donor.

1. Type the SMARTS string into the SMARTS pattern text box, or click Get From Selection to use the selected atoms in the Workspace to define a SMARTS string.

   If you use the Workspace selection to define the SMARTS pattern, you might want to edit it before proceeding.

2. Choose Group, Point, or Vector from the Geometry option menu.

   The remaining controls in the dialog box depend on the choice you make from this menu.

   - Group—The pattern contributes a group of atoms to the pharmacophore feature definition, with the pharmacophore site placed at the centroid. The Projected point type menu has only none and aromatic ring options available, and the Group atoms controls are displayed.
   - Point—The pattern contributes a single atom to the pharmacophore feature definition, with the pharmacophore site placed at that atom. The only available item on the Projected point type menu is none, and the Point atom text box is displayed.
   - Vector—The pattern contributes an atom with one or more directions to the pharmacophore feature definition, with the pharmacophore site placed at the atom. The Projected point type menu has items for donor and acceptor groups, and the Vector atom text box is displayed.

3. Choose the point type from the Projected point type option menu:

   - Group: Choose aromatic ring if the SMARTS pattern defines an aromatic ring, otherwise choose none.
   - Point: none is the only available choice.
   - Vector: Choose donor if the pattern represents a hydrogen bond donor, or choose the acceptor, sp*n*, *m* lp item that defines the type of acceptor (hybridization and number of lone pairs at the acceptor) if the pattern represents a hydrogen bond acceptor.

4. Choose the atoms that define the pharmacophore site:

   - Group: Select All if all atoms in the SMARTS pattern are to be used to define the group centroid, or select Numbers and type the atom numbers for the group centroid in the text box, separated by commas.

- Point: Type the atom number for the pharmacophore site in the Point atom text box.
- Vector: Type the atom number for the pharmacophore site in the Vector atom text box. This should be the donor or acceptor atom.

The atom numbers refer to the order of the atoms in the SMARTS string.

Once you have added a pattern, you can edit it by clicking Edit. The Edit Pattern dialog box is displayed. This dialog box has the same controls as the New Pattern dialog box. If you no longer need the pattern, you can click Delete to delete it. However, you can also ignore it, if you want to keep it in the definition for other applications, but not use it—see the next section. Both of these buttons are only available when you select a custom pattern. Custom patterns are highlighted in blue in the Pattern list table.

5.2.3 Choosing How Patterns Are Used

Matching of patterns to ligand structures is done in the order specified in the Pattern list table. For example, if the first pattern maps a particular nitrogen in the ligand as an acceptor, that same nitrogen will not be mapped as an acceptor by any subsequent pattern. If you have added custom patterns, you can move them up and down the list with the arrow buttons below the table to set their priority. You cannot change the order of the built-in patterns.

If you want to exclude functional groups represented by a pattern from the feature, you can select the check box in the Exclude column for the pattern. For example, you might want to exclude a carboxylic acid group from being considered as a hydrogen bond donor, because it will be ionized under physiological conditions. Excluded functional groups are processed before included groups, so their position in the table does not matter.

If you want a pattern to be ignored, you can select the check box in the Ignore column. Ignored patterns are equivalent to deleted patterns. If you want to save a custom pattern for later use, but not use it in the current feature, select the check box in the Ignore column.

5.2.4 Viewing Patterns

The patterns that define a feature can be viewed individually in the Workspace for each ligand. To display a pattern for a particular ligand, select the ligand in the Ligands table (in the Define Pharmacophore Model panel), then select the check box in the Mark column of the Pattern list table (in the Edit Features dialog box) for the desired pattern. Any occurrences of the pattern are marked in the ligand structure.

You can display markers for more than one pattern, but the markers do not distinguish between patterns. You can display markers for more than one ligand by including the ligands in the Workspace. To see the atoms and bonds as well as the markers, select Apply marker offset.

5.2.5 Adding Custom Features

Phase allows you to define up to three custom features. To add a feature, click Add Custom Feature. The Add Custom Feature dialog box is displayed, in which you can specify a name and choose a code letter for the new custom feature. The custom feature is added to the Feature option menu and selected, and the Pattern list table is cleared. You can now add patterns to the feature and set their status as described in the sections above.

If you no longer need the feature, you can click Delete Custom Feature to delete it.

5.2.6 Using Projected Points

By default, donors and acceptors are represented by vectors originating at the donor (hydrogen) or acceptor atom. The alignment of these vectors is used to determine whether ligands share the associated feature. Sometimes, two active ligands can form a hydrogen bond to the same receptor site, but from different directions. The projected point is in the same location but the ligand features are not. With the default representation, these two ligands would not contribute to the same pharmacophore hypothesis.

You can replace the vectors with points at a specified distance from the ligand donor or acceptor atom. These points simulate the corresponding acceptor or donor in the receptor, and are called *projected points*. In the default feature set, the projected points are implicit. To use projected points, select Projected points only, and enter a distance in angstroms in the Distance text box. With this option, only the patterns that have a vector geometry and a defined projected point type contribute to the feature. All other patterns that are not excluded are ignored. Vector alignments are not used because the vectors have been replaced by points.

5.3 Defining the Ligand Set for Model Development

If you have not already done so, you can define the active and inactive ligands that are used to develop the pharmacophore model in this step. The controls for doing so are the same as in the Prepare Ligands step. You can click Activity Thresholds to set thresholds for the activity, or you can select ligands for the active set in the Pharm Set column of the Ligands table. See Section 4.5 on page 48 for more information.

5.4 Creating the Sites

Once you are satisfied with the feature set, click Create Sites to start the job that creates and stores the site points for each conformer of each ligand. If the sites already exist for a conformer set (because you copied them from another run, for example), a link is made to this set instead of running the job.

5.5 Step Summary

To create site points for each ligand:

1. Click Create Sites.
2. Click Next to proceed to the next step.

Optional tasks:

- Add to the existing features, create custom features, and exclude or ignore patterns by clicking Edit Features.
- Select the use of projected points for acceptors and donors rather than treating them as vector features.
- Define the active and inactive ligands by clicking Activity Thresholds or clicking in the Pharm Set column of the Ligands table.

Finding Common Pharmacophores

In the Find Common Pharmacophores step, pharmacophores from all conformations of the ligands in the active set are examined, and those pharmacophores that contain identical sets of features with very similar spatial arrangements are grouped together. If a given group is found to contain at least one pharmacophore from each ligand, then this group gives rise to a *common pharmacophore*. Any single pharmacophore in the group could ultimately become a common pharmacophore *hypothesis*—an explanation of how ligands bind to the receptor.

Common pharmacophores are identified from a set of *variants*. A variant is a set of feature types that define a possible pharmacophore—for example, the variant ADHH contains a hydrogen-bond acceptor, a hydrogen-bond donor, and two hydrophobic groups.

Phase searches for common pharmacophores with a given number of pharmacophore sites. You can specify from 3 to 7 sites: hypotheses with more sites are not likely because each site represents a 2-3 kcal/mol interaction with the receptor. In addition, you can control how many ligands must match to form a valid hypothesis, and how many of each kind of feature must be included in the match. After the search is complete, the variants for which common pharmacophores were found are passed to the next step.

6.1 The Search Method

Common pharmacophores are identified using a tree-based partitioning technique that groups together similar pharmacophores according to their *intersite distances*, i.e., the distances between pairs of sites in the pharmacophore. Each k-point pharmacophore is represented by a vector of n distances, where $n = k \cdot (k-1)/2$. Each intersite distance d is filtered through a binary decision tree, such as in Figure 6.1.

The tree in Figure 6.1 has a depth of four and partitions distances (in angstroms) on the interval (0, 16] into bins that are 2 Å wide. If each of the n distances in a pharmacophore is filtered in this manner, an n-dimensional partitioning of the pharmacophore is created. This representation is referred to as an n-dimensional box, where the sides of the box are equal to the bin width. Thus a pharmacophore is mapped, according to its intersite distances, into a box of finite size. All pharmacophores that are mapped into the same box are considered to be similar enough to facilitate identification of a common pharmacophore. So if each of the minimum required number of active-set ligands contributes at least one pharmacophore to a particular box, then that box represents a common pharmacophore. Such boxes are said to *survive* the partitioning procedure, while all others are eliminated.

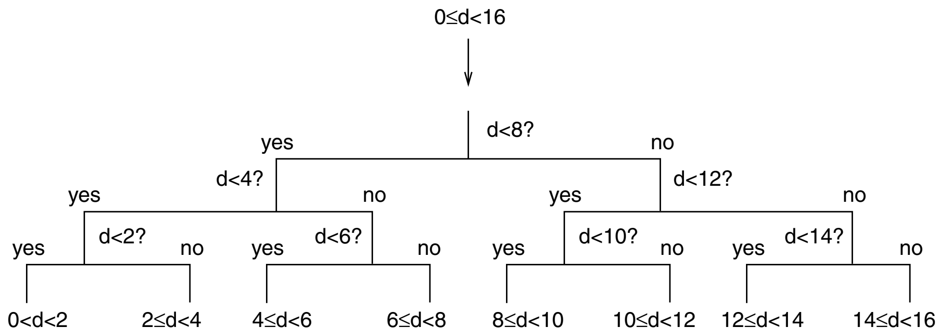


Figure 6.1. Binary decision tree.

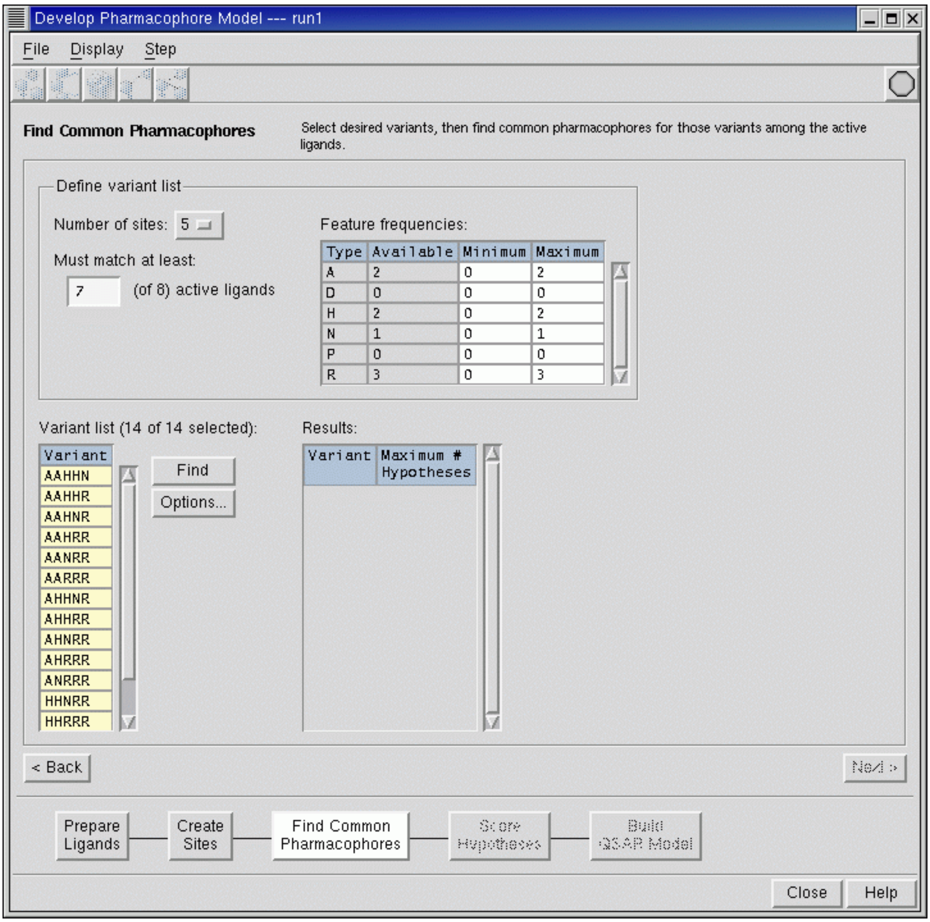


Figure 6.2. The Find Common Pharmacophores ***step.***

6.2 Defining the Scope of the Search

Searching for all possible common pharmacophores could take a long time. From your knowledge of the system of interest, you might not want to search for pharmacophores that have too many or too few site points, or that have too many or too few features of a particular type. Phase provides the means to narrow the search to the variants of interest.

When you enter this step for the first time, a list of all available variants in the set of ligands designated as active is computed, from the number of available sites of each type for each ligand. This list is usually shorter than the theoretical maximum length, because the ligands don't necessarily include all possible variants. The list is filtered with the default settings for the number of sites before it is displayed in the Variant list table. The frequencies of occurrence of the features are used to determine how many occurrences of each feature could be found in a valid hypothesis, given the number of ligands that must be matched. These values are listed in the Available column of the Feature frequencies table, which is described in Table 6.1.

The first task is to decide how many site points to include in the hypothesis. The default is 5, but you can choose any number between 3 and 7, inclusive, from the Number of sites option menu. When you make your choice, you should be aware that the likelihood of finding common pharmacophores is decreased as the number of sites is increased.

If you want to examine hypotheses with different numbers of points, you can create a new run for each number of points. To create a run that stores the information to date, choose Save As from the File menu, and name the new run. The original run is preserved, and you are now working in the new run. To revert to the original run, choose the run from the Open submenu of the File menu.

Table 6.1. Description of Feature frequencies table columns.

Column	Description
Type	Lists the features by code letter. Noneditable.
Available	Number of sites of this type that are available, defined as the largest number of occurrences of this feature for which a match can be found for the number of ligands to be matched. For example, with ten ligands, of which three ligands have 4 Acceptors, six have 5 Acceptors, and one has 6 Acceptors, the number of Acceptors available is 4 if all ten ligands are matched, but is 5 if seven ligands are matched. Noneditable; updated if the number of ligands to match changes.
Minimum	Minimum number of features of the given type allowed in any variant. Editable: you can set the value to restrict the possible variants. The default is zero.
Maximum	Maximum number of features of the given type allowed in any variant. Editable: you can set the value to restrict the possible variants. The default is the maximum possible number, given the number available and the number of sites.

By default, all of the active-set ligands must contain a given variant for that variant to be listed. However, Phase allows you to relax this criterion so that a common pharmacophore need only match a subset of the chosen actives. This is often a necessity when more than one binding mode is observed among the actives. If you want to widen the search, you can set the number of actives that must contain the variant to a number less than the total, in the At least text box. The number must be between 1 and the maximum, inclusive. The maximum number (the number of chosen actives) is displayed to the right of the text box. The fewer ligands you require a match to, the more variants will be listed.

Not all the variants are likely to be useful: for example, a variant with five acceptors might be physically unreasonable, and should be excluded from the search. You can limit the number of occurrences of any of the features by entering a minimum and maximum number in the Minimum and Maximum columns of the Feature frequencies table. For example, you might want variants that have between 1 and 3 acceptors. In this case you would enter 1 in the Minimum column of the A row, and 3 in the Minimum column.

After each change, the variant list is automatically updated.

6.3 Modifying the Search Parameters

In the Find Common Pharmacophores - Options dialog box, you can specify the parameters that govern the search for common pharmacophores. Specifying the parameters is a balance between the size of the features in the hypotheses, the size of the ligands, and the time taken and storage requirements for the search. To open this dialog box, click Options in the Find Common Pharmacophores step. The text boxes are described below.

Minimum intersite distance

Specifies the minimum distance allowed between two features. If the features in the ligand are closer than this distance, the hypothesis is rejected.

Maximum tree depth

Specifies the number of binary partitioning steps used to sort the pharmacophores into similar groups. This is the maximum recursion level in the partitioning, and the depth of the resulting binary partitioning tree.

Initial box size

Noneditable. Specifies the size of the initial box in the partitioning algorithm, computed from the final box size and the maximum tree depth:

$$\text{Initial box size} = (\text{Final box size})^* \, 2^{(\text{Maximum tree depth})}.$$

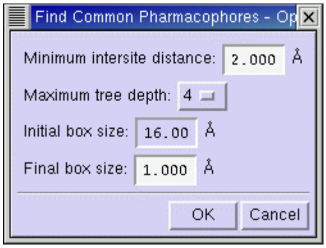


Figure 6.3. The Find Common Pharmacophores - Options ***dialog box.***

This size should be roughly the size of the binding pocket, or of the smallest ligand. You should therefore choose the final box size and the maximum tree depth to ensure that the initial box size is big enough.

Final box size

Specifies the size of the boxes that contain intersite distances that are considered to be equivalent. This option governs the tolerance on matching: the smaller the box size, the more closely pharmacophores must match. However, the smaller the box size, the longer the search takes. If you choose a smaller final box size, you might have to increase the maximum tree depth so that the initial box size is large enough. If the final box size is too small, the tolerance on matching might be too strict to produce any common pharmacophores.

6.4 Starting the Search

When you have defined the list of variants, you can proceed to the search for common pharmacophores. The search is performed on the variants that are selected in the Variant list table. By default, all variants are selected. You can select variants from the Variant list using the usual combinations of click, control-click, and shift-click. You must have at least one variant selected to run the search.

The common pharmacophores are identified using a binary partitioning algorithm, in which the pharmacophores are split into progressively smaller and more similar groups based on intersite distances. If you want to change the parameters of the search, you can do so by clicking Options and setting the values in the dialog box that is displayed—see Section 6.3 on page 62.

To start the search, click Find. A dialog box is displayed, in which you can select the host, the number of CPUs to use, and the user name on the host. You can distribute this job over multiple processors.

The results of the search can take a large amount of disk space, depending on the number of ligands and their size and flexibility. The search results are kept inside the run, which is stored in the Maestro project. You should make sure that you have adequate disk space: in the temporary storage on the host or hosts on which you run the job, in the Maestro I/O directory, and in the project. Because of the disk space requirements, it is not advisable to run from a scratch project, which is kept by default in `/home/`*username*`/.schrodinger`. Instead, you should save the project to a disk that has plenty of free space.

The results of the run are displayed in the Results table. This table shows the maximum number of hypotheses that could be produced by each variant. You can sort the table by clicking the column headers. Some variants may have no common pharmacophores. These variants are not passed to the next step.

6.5 Step Summary

To find common pharmacophores:

1. Choose the number of sites from the Number of sites option menu.
2. Specify the number of actives to match in the Must match section.
3. Set limits on the minimum and maximum number of features of each type in the Feature frequencies table.
4. Select variants from the Variant list.
5. (Optional) Set search parameters by clicking Options and entering values in the Find Common Pharmacophores - Options dialog box.
6. Start the search by clicking Find.
7. Click Next to proceed to the next step.

Scoring Hypotheses

In the Score Hypotheses step, common pharmacophores are examined, and a scoring procedure is applied to identify the pharmacophore from each surviving *n*-dimensional box that yields the best alignment of the chosen actives. This pharmacophore provides a hypothesis to explain how the active molecules bind to the receptor. There will of course be many hypotheses, because there are many boxes. The scoring procedure provides a ranking of the different hypotheses, allowing you to make rational choices about which hypotheses are most appropriate for further investigation.

Following the scoring of the hypotheses, the remaining molecules can be used to provide extra information in the hypothesis, based on their structure. To make comparisons, Phase uses *partial matching* to obtain alignments for these ligands. If at least three sites in the hypothesis can be matched, an unambiguous alignment is obtained. For each ligand not designated active, Phase searches for matches involving the largest possible number of sites, and identifies the match that yields the highest fitness score.

If the pharmacophore is an adequate hypothesis, it should discriminate between active and inactive molecules. By aligning and scoring known inactives, you can check the validity of the hypotheses that you generated. If inactives score well, the hypothesis could be invalid because it does not discriminate between actives and inactives, and therefore does not explain how active molecules bind but inactives do not. The hypothesis could also be incomplete because it lacks either a critical site that explains the binding or information on what prevents inactives from binding.

The pharmacophore features that were identified are not the only features that may be useful in defining a good hypothesis. Inactive molecules that have the same pharmacophore features could have functional groups in regions of space not occupied by the active molecules. It is reasonable to suppose that these regions are occupied by the receptor. These regions can then be added to the hypothesis as *excluded volumes*, and used in the database search to screen matches to the hypothesis.

Inactive molecules could also have different functional groups in the same location as functional groups in the active molecules, or be missing functional groups that are in the active molecules. Visual inspection of the aligned ligands can help you understand the structural differences. These can also be quantified by building a QSAR model (in the next step), which can be used for screening matches in the database search as well as for identifying functional groups that contribute, positively or negatively, to activity.

7.1 The Scoring Process

A surviving box contains a set of very similar pharmacophores culled from conformations of a minimum number of active-set ligands, and certain of these ligands may contribute more than one pharmacophore to a box. Each pharmacophore and its associated ligand are treated temporarily as a *reference* in order to assign a score. This means the other *non-reference* pharmacophores in the box are aligned, one-by-one, to the reference pharmacophore, using a standard least-squares procedure applied to the corresponding pairs of site points.

During this process, the quality of each alignment is measured in three ways: (1) the *alignment score*, which is the root-mean-squared deviation (RMSD) in the site-point positions; (2) the *vector score*, which is the average cosine of the angles formed by corresponding pairs of vector features (acceptors, donors, and aromatic rings) in the aligned structures; and (3) a *volume score* based on the overlap of van der Waals models of the non-hydrogen atoms in each pair of structures.

$$S_{\mathrm{vol}}(i) = V_{\mathrm{common}}(i)/V_{\mathrm{total}}(i)$$

$V_{common}(i)$ is the common or overlapping volume between ligand i and the reference ligand, while $V_{total}(i)$ is the total volume occupied by both ligands.

In principle, a reference pharmacophore could score well, even though it contains one or two very poor individual alignments. For this reason, user-adjustable cutoffs are applied to the RMSD values and vector cosines of each individual alignment. Any reference pharmacophore that violates a cutoff in any individual alignment is eliminated. A *site score* for each alignment is then computed based on the alignment score $S_{\mathrm{align}}(i)$ and the cutoff C_{align} by

$$S_{\mathrm{site}}(i) = 1 - S_{\mathrm{align}}(i)/C_{\mathrm{align}} .$$

This score is always between 0 and 1 because alignments with $S_{\mathrm{align}}(i) > C_{\mathrm{align}}$ are eliminated.

The site score, the vector score, and the volume score are combined with separate weights to yield a combined alignment score for each non-reference pharmacophore that has been aligned to the reference. If a non-reference ligand contributes more than one pharmacophore to the box, the pharmacophore yielding the best alignment to the reference is selected. The overall multi-ligand alignment score for a given reference pharmacophore is the average score from the best individual alignments.

After all pharmacophores in a box have been treated as a reference, the one yielding the highest multi-ligand alignment score is selected as the hypothesis for that box. The ligand that contributes the reference pharmacophore is referred to as the *reference ligand* for that hypothesis. The non-reference information is carried along with each hypothesis so that additional scoring can be performed using the optimal multi-ligand alignment.

Once hypotheses have been identified across all boxes, the lower scoring hypotheses can be eliminated by applying a percentage cutoff to the overall alignment score. In case the percentage filter yields a very small number of hypotheses, a minimum number of hypotheses can be specified.

After this stage of scoring is completed, the ranking of the hypotheses can be refined using volume and selectivity scoring. The overall volume score for a hypothesis is the average obtained by applying the formula given above to all non-reference ligands i. The volume score (S_{vol}) can be added to the overall score with its own user-adjustable weight (W_{vol}).

Selectivity is an empirical estimate of the *rarity* of a hypothesis, i.e., what fraction of molecules are likely to match the hypothesis, regardless of their activity toward the receptor. Selectivity is defined on a logarithmic scale, so a value of 2 means that 1 in 10^2 molecules would be expected to match the hypothesis. Higher selectivity is desirable because it indicates that the hypothesis is more likely to be unique to the active-set ligands. Selectivity is only a rough estimate of the rarity, so you should be careful not to place too much emphasis on it in the overall ranking of hypotheses. As with the other types of scores, the selectivity score (S_{sel}) can be added to the overall score with its own user-adjustable weight (W_{sel}).

If you choose to match less than the total number of chosen actives, you may wish to assign higher scores to hypotheses that match a greater number of the chosen actives. The reward comes in the form of W_{rew}^{m} , where W_{rew} is user-adjustable (1.0 by default) and m is the number of actives that match the hypothesis minus one. If W_{rew} is increased much above 1.0, care must be taken not to make it too large, or it may completely dominate the scoring function. For example, if you have 10 actives and W_{rew} is 1.4, this contribution to the score could have a value of 32. The other terms have a maximum value of 1.0.

Hypotheses for which the reference ligand has a high energy relative to the lowest-energy conformer for that ligand are less likely to be good models of binding, because of the energetic cost. You can include a penalty for high-energy structures by subtracting a multiple of the relative energy from the final score, $W_{\mathrm{E}}\,\Delta E$.

Likewise, you can penalize hypotheses for which the reference ligand activity is lower than the highest activity, by adding a multiple of the reference ligand activity to the score, $W_{\mathrm{act}}\,A$, where A is the activity.

The final scoring function—the *survival score*—has the following form:

$$S = W_{\mathrm{site}} S_{\mathrm{site}} + W_{\mathrm{vec}}\, S_{\mathrm{vec}} + W_{\mathrm{vol}}\, S_{\mathrm{vol}} + W_{\mathrm{sel}}\, S_{\mathrm{sel}} + W_{\mathrm{rew}}^{m} - W_{\mathrm{E}} \Delta E + W_{\mathrm{act}}\, A$$

where the W's are the weights and the S's the scores.

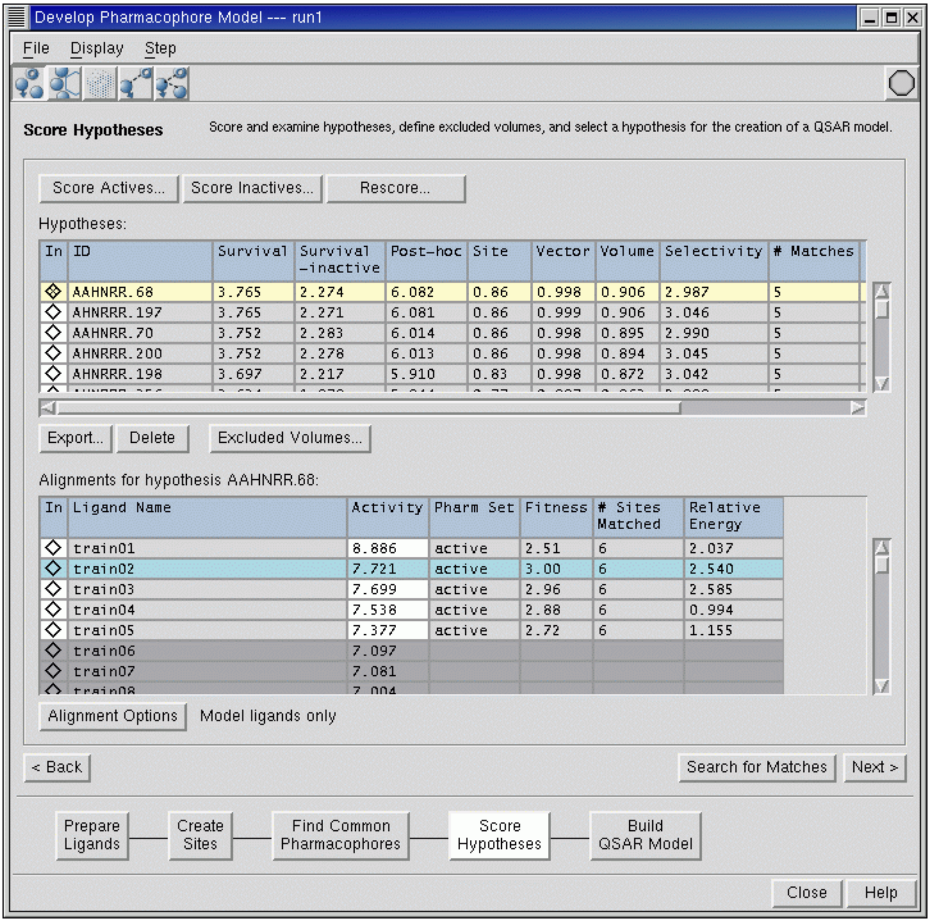


Figure 7.1. The Score Hypotheses ***step, after scoring.***

7.2 Scoring the Hypotheses

The first task in the Score Hypotheses step (Figure 7.1) is to align the actives to the hypotheses and calculate the score for the actives. To do this, click Score Actives. When you do so, the Score Actives dialog box (Figure 7.3) is displayed, in which you can examine and adjust the weights of the terms in the survival score, the alignment thresholds, and the filters on the number of hypotheses to keep. Details of these parameters are given in the sections below.

When you have made any changes, click Start, set any job parameters in the Start dialog box, and click Start again. When the job finishes, the surviving hypotheses and their scores are displayed in the Hypotheses table.

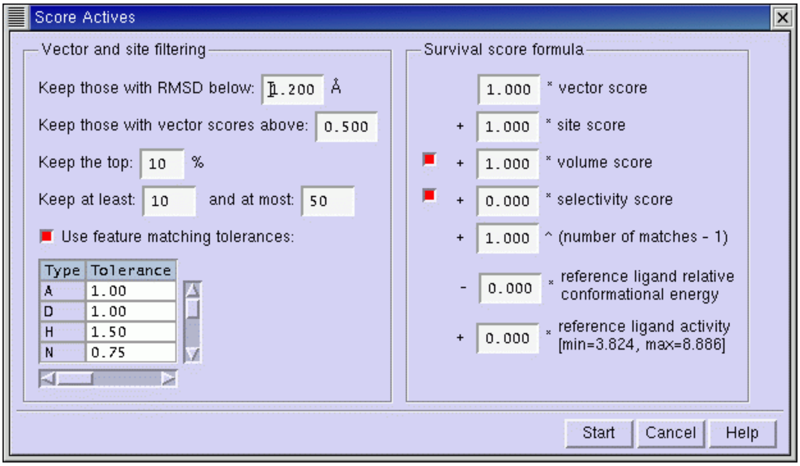


Figure 7.2. The Score Actives ***dialog box.***

7.2.1 Scoring Method and Filtering

In this section, you can set the thresholds for filtering out hypotheses with low alignment scores and with poor feature matching, and set a limit on the number of hypotheses to keep.

Alignment Scores

Vector and site alignment scores are computed first, and used to filter the hypotheses. You can set the following parameters, all of which are applied to filter the hypotheses:

Keep those with RMSD below *threshold* Å

Threshold for RMS deviation of the intersite distances of any contributing ligand from those of the reference ligand. The default is 1.2 Å.

Keep those with vector scores above *threshold*

Threshold for the variation in the alignment of vectors between any contributing ligand and the reference ligand. The maximum is 1.0, which corresponds to perfect alignment. The minimum is –1.0, which would keep all hypotheses, regardless of vector alignment.

Numerical Cutoffs

To limit the number of hypotheses, you can set the following cutoffs on the fraction or number of hypotheses to keep.

Keep the top *n* %

Limit on the percentage of hypotheses to keep, in order of combined alignment score.

Keep at least *n* and at most *m*

Lower and upper limits for the number of hypotheses to keep. If the percentage of hypotheses kept is lower or higher than these limits, these limits override the percentage limit.

Feature-Matching Tolerances

In addition to using the RMSD to filter out hypotheses, you can set matching tolerances on individual features. Features are considered to match if the site points are within the specified tolerance. This feature is useful if the RMSD matching is satisfied, but one or more features does not match well enough.

To apply feature-matching tolerances, select Use feature matching tolerances. The tolerances for each feature type are listed in the table below, and can be edited. All tolerances are applied: if you want to disable matching for a particular feature type, set the tolerance to a large value.

7.2.2 Survival Score Weighting Factors

The Weighting factors section of the Score Actives dialog box defines the survival score of the hypotheses, which is reported in the Hypotheses table of the Score Hypotheses step along with the individual scores that make up the survival score. The possible ranges for each score and weight are given in Table 7.1.

Table 7.1. Maximum ranges for score components and allowed weight ranges in the survival score

Score	Score Range	Weight Range	Default Weight
vector score	–1.0 to 1.0	0.0 to 1.0	1.0
site score	0.0 to 1.0	0.0 to 1.0	1.0
volume score	0.0 to 1.0	0.0 to 1.0	1.0
selectivity score	0.0 to ∞	0.0 to 1.0	0.0
number of matches	1.0 to ∞	1.0 to ∞	1.0
reference ligand relative conformational energy	0.0 to ∞	0.0 to ∞	0.0
reference ligand activity	Determined by input	0.0 to ∞	0.0

The lower end of the actual range for the vector score is limited by the cutoff specified in the Vector and site filtering section. Similarly, the maximum relative energy is limited by any cutoff you specified when generating conformers, such as in the Generate Conformers dialog box (see Section 4.3 on page 43).

The selectivity score weight is zero by default because it might eliminate useful hypotheses. Likewise, the energy and activity weights are zero by default.

The weight for the number of matches is raised to the power of the number of matches minus one. A value of 1.0 does not discriminate on the basis of the number of matches. This score can be useful when the required minimum number of actives is smaller than the total number of actives. Adjust this weight with caution: even a value of 2.0 can give large variations in survival scores, and dominate the survival score.

7.3 Scoring Inactives and Rescoring

Once the hypotheses have been scored on the basis of the alignment of the chosen actives, you can calculate an adjusted score based on the alignment of the chosen inactives. The score is adjusted by subtracting a multiple of the survival score of the inactives from the survival score of the actives. To calculate this score, click Score Inactives, specify a weight for the inactive score, and click Start. After making any job settings in the Start dialog box, click Start again. When the job finishes, the adjusted scores are displayed in the Survival-inactive column of the Hypotheses table.

If you want to apply a different scoring function to the surviving hypotheses, you can do so by clicking Rescore, and setting values for the coefficients (weights) of the scoring function in the Rescore Hypotheses dialog box (Figure 7.4). This dialog box contains the same controls as in the Score Actives dialog box. At the same time, you can adjust the weight of the inactives in the Survival-inactive score. The results of the rescoring are listed in the Post-hoc column of the Hypotheses table. These results correspond to the Survival score, and do not include any penalty for matching inactives.

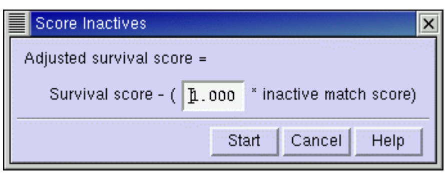


Figure 7.3. The Score Inactives ***dialog box.***

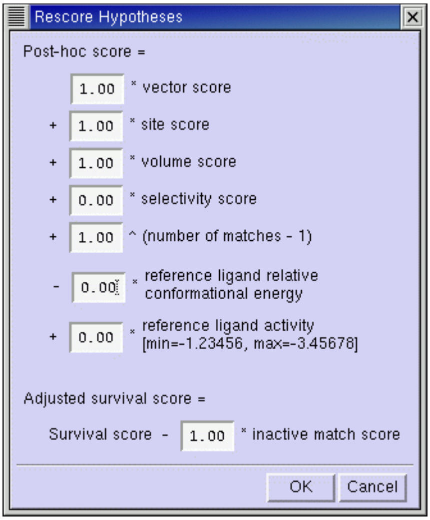


Figure 7.4. The Rescore Hypotheses ***dialog box.***

7.4 Results of Scoring

The Hypotheses table displays the scores for each hypothesis. The hypothesis ID is given in the first column, and consists of the variant name and an index. The remaining columns contain the various scores, whose definitions are given in Table 7.2. You can sort the table by the values in a column by clicking the column heading. The data in the table is noneditable.

Table 7.2. Description of score columns in the Hypothesis table.

Column	Description
Survival	Weighted combination of the vector, site, volume, and survival scores, and a term for the number of matches. The weights of the volume score and survival score are set to 1.0 and 0.0 by default. The weights can be varied in the Score Actives dialog box. The minimum value of this score is 1.0.
Survival - inactives	Survival score calculated for actives with a multiple of the survival score calculated for inactives subtracted. The weight of the inactive survival score can be set in the Score Inactives dialog box.
Post-hoc	This score is the result of rescoring, and is a weighted combination of the vector, site, volume, and selectivity scores. You can set the weights in the Rescore Hypotheses dialog box, which you open by clicking Rescore.
Site	Site score. This score measures how closely the site points are superimposed in an alignment to the pharmacophore of the structures that contribute to this hypothesis, based on the RMS deviation of the site points of a ligand from those of the reference ligand.

Table 7.2. Description of score columns in the Hypothesis table. (Continued)

Column	Description
Vector	Vector alignment score. This score measures how well the vectors for acceptors, donors, and aromatic rings are aligned in the structures that contribute to this hypothesis, when the structures themselves are aligned to the pharmacophore. The vector score is the average of the cosines of the angles between the vectors of each aligned ligand and those of the reference ligand.
Volume	Measures how much the volumes of the contributing structures overlap when aligned on the pharmacophore. The volume score is the average of the individual volume scores. The individual volume score is the overlap of the volume of an aligned ligand with that of the reference ligand, divided by the total volume occupied by the two ligands.
Selectivity	Estimate of the rarity of the hypothesis, based on the World Drug Index. The selectivity is the negative logarithm of the fraction of molecules in the Index that match the hypothesis. A selectivity of 2 means that 1 in 100 molecules match. High selectivity means that the hypothesis is more likely to be unique to the actives.
# Matches	Number of actives that match the hypothesis.
Energy	Relative energy of the reference ligand in kcal/mol. This is the energy of the reference conformation relative to the lowest-energy conformation.
Activity	Activity of the reference ligand.
Inactive	Survival score of inactives. The scoring function is the same as for actives.

7.5 Examining Hypotheses and Ligand Alignments

Once you have generated scores for the hypotheses, you can examine the listed hypotheses, one at a time. To examine a hypothesis, select it in the Hypotheses table. The ligands that fit this hypothesis are listed in the Alignments table. The first four columns contain the same information or controls as the Ligands table in the Prepare Ligands step. The columns of this table are described in Table 7.3.

The columns of this table are noneditable. The row for the reference ligand (the ligand that matches the hypothesis exactly) is colored light blue. Rows for unaligned ligands are colored dark gray. You can sort the table by the values in a column by clicking the column heading.

You can view the aligned ligands and information on their alignments in the Workspace by clicking the diamond in the In column of the Alignments table. This column is multiple-select: you can add ligands to the display with shift-click and control-click. You can display and undisplay the hypothesis by clicking the View Hypothesis toolbar button. From the toolbar (or the Display menu) you can also display the distances between the site points and the angles between all sets of three site points.

Table 7.3. Description of Alignments table.

Column	Description
In	Inclusion status of the ligand. The diamond has a cross in it if the ligand is included in the Workspace, and is empty if the ligand is excluded. You can include and exclude ligands with click, shift-click, and control-click.
Ligand Name	The name of the ligand.
Activity	The ligand's activity. This value is editable.
Pharm Set	Indicates the status of a ligand in the set used to bind the pharmacophore model.
Fitness	Measures how well the conformer matches the hypothesis. The fitness score is a linear combination of the site and vector alignment scores and the volume score, and is related to the default survival score. The reference ligand, which matches exactly, has a perfect fitness score.
# Sites Matched	Number of sites on the ligand that matched the hypothesis.
Relative Energy	Energy of the best matching conformer relative to the lowest conformer. High relative energies indicate that the conformer is strained. A large proportion of high relative energies could indicate a poor hypothesis.

It can also be useful to align and display the "non-model" molecules: the inactive molecules from the pharm set, the actives that do not match all site points, and the molecules that are not in the pharm set. To align these ligands, click Alignment Options, below the Alignments table. In the Alignment Options dialog box (Figure 7.5) select Align non-model ligands. By default the molecules can match any three sites, but you can enforce matching at specific sites by selecting the sites under Must match. The tolerances for matching these sites are the tolerances specified in the Score Actives dialog box. When you have made your selections, click OK. The dialog box closes and the alignment is performed. The table is updated with information for all molecules that match three or more sites in the hypothesis. The rows for molecules that have not been aligned are colored dark gray and are missing the information coming from the alignment.

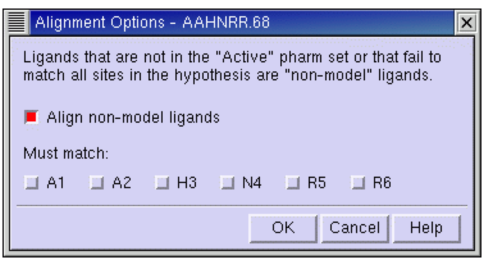


Figure 7.5. The Alignment Options ***dialog box.***

If, in your examination of a hypothesis, you decide that the hypothesis is not a good one, you can delete it by clicking Delete. If you have found a hypothesis that you want to use outside the project—for example, to search a database without opening the project—you can export the hypothesis to a file by clicking Export, navigating to the desired location and providing the file name. Hypotheses are stored in the run as part of the project, so they can always be used inside the project, but they must be exported to be used for database searching in some other project.

7.6 Adding Excluded Volumes to Hypotheses

If you have included inactive molecules in the Phase run, you can use these molecules to add regions of space to the hypothesis in which there should be no atoms from any active molecule. These excluded volumes are then used when you search for matches in your database to screen out ligands that have atoms in the excluded volumes.

To define one or more excluded volumes using the inactive molecules as a guide, first make sure that you have aligned the non-model ligands, then display one or more of the active molecules along with one or more of the inactive molecules. Regions of space in which inactive molecules have atoms but active molecules do not are likely candidates for excluded volumes.

With the ligands displayed, click Excluded Volumes. The Excluded Volumes dialog box is displayed. This dialog box allows you to pick atoms to define spherical excluded volumes.

To define a volume, pick a set of atoms in the Workspace that belongs to the inactive molecule or molecules. After each pick, the centroid of the atom set is calculated and a temporary sphere of radius 1 Å is displayed at the centroid. If you pick an already picked atom, it is removed from the set, the centroid is recomputed, and the sphere is moved.

When you have picked an appropriate set of atoms, click Add Volume to add the sphere to the hypothesis as an excluded volume. Once the sphere is added, it is no longer connected in any way with the atoms that you used to define it. You can subsequently change the radius of the sphere and its location by editing the table cells.

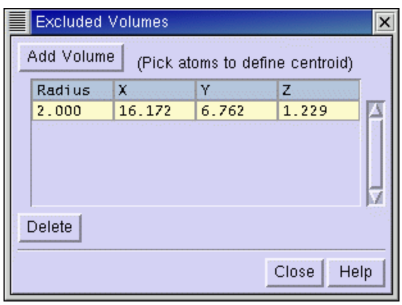


Figure 7.6. The Excluded Volumes ***dialog box.***

To display excluded volumes, select them in the table. You can select multiple rows in the table, with shift-click and control-click. The spheres for the selected rows are highlighted in the Workspace. After you have dismissed the Excluded Volumes dialog box, you can view the excluded volumes in the Workspace using the toolbar button or from the Display menu.

To delete excluded volumes, select them in the table and click Delete.

7.7 Step Summary

To score hypotheses:

1. Click Score Actives.
2. Set scoring options in the Score Actives dialog box.
3. Score the hypotheses by clicking OK.

Optional tasks:

- Score inactives to generate an adjusted scoring function by clicking Score Inactives.
- Rescore the hypotheses with an adjusted scoring function by clicking Rescore.
- Export the selected hypothesis to a file, by clicking Export.
- Add excluded volumes to the selected hypothesis, by clicking Excluded Volumes.
- View hypotheses and alignments in the Workspace, using the toolbar buttons and the Alignments table.

To proceed to building QSAR models:

1. Select the desired hypotheses in the Hypotheses table.
2. Click Next.

To proceed to searching for matches:

1. Select the desired hypotheses in the Hypotheses table.
2. Click Search for Matches.

Building a QSAR Model

Phase provides the means to build 3D QSAR models for a set of ligands that are aligned to a selection of hypotheses, and to visualize these models along with the ligand structures and the hypotheses. The QSAR models are developed from a series of ligands that have a range of activities. The usefulness of the QSAR model depends on how well the activity range is spanned, and how diverse the structures are.

In the Build QSAR Model step, you build QSAR models for the hypotheses selected in the Score Hypotheses step, using the activity data for all the available ligands. You can choose atom-based or pharmacophore-based models, select different training sets and test sets, vary the grid spacing, and visualize the model results. When you have built the models, you can use them to visualize parts of the ligands (atoms or pharmacophores) that contribute positively or negatively to activity, and to predict activities of matches to the hypotheses from a database.

When you have completed this step, you can export the hypotheses used to build the model to an external file for use with other projects, and you can continue directly to a search for matches to the hypotheses.

8.1 Phase QSAR Models

Phase QSAR models are 3D QSAR models, in which chemical features of ligand structures are mapped to a cubic 3D grid. The ligands are first aligned to the set of pharmacophore features in the selected hypotheses using a standard least-squares procedure, as outlined in Section 7.1 on page 66. A rectangular grid is defined to encompass the space occupied by the aligned ligands. This grid divides the occupied space into *N* uniformly-sized cubes, typically 1 Å on each side.

The independent variables in the regression are the binary-valued occupancies of the cubes by structural components; the dependent variables are the activities. The regression is performed by a partial least squares (PLS) method, in which a series of models is constructed with an increasing number of PLS factors. The accuracy of the models increases with increasing number of PLS factors until over-fitting starts to occur.

Phase offers two choices for the structural components that form the basis of the model: atoms and pharmacophore features.

In the atom-based QSAR models, the structural components of the ligands are represented by van der Waals models of the atoms in the ligands. Each atom is treated as a sphere whose radius is the van der Waals radius for the MacroModel atom type. To distinguish different atom types that occupy the same regions of space, atoms are divided into six classes:

- D—Hydrogen-bond donor (hydrogens bonded to N, O, P, S)
- H—Hydrophobic or nonpolar (C, H–C, Cl, Br, F, I)
- N—Negative ionic (formal negative charge)
- P—Positive ionic (formal positive charge)
- W—Electron-withdrawing (N, O; includes hydrogen-bond acceptors)
- X—Miscellaneous (all other types)

These classes have some correspondence to pharmacophore feature types, but atom classes are assigned using fixed internal rules, not the hypothesis feature definitions.

A given atom can occupy the space of one or more cubes in the grid. A cube is occupied by an atom of a particular class if the center of that cube falls within the radius of the atom. Each ligand can therefore be represented by a set of bit values (0 or 1) that indicate which cubes are occupied by atoms of each class. The independent variables used in the QSAR model are the $6N$ occupancies of the cubes and atom classes: each variable corresponds to a given cube and a given atom class, and can take the value 0 or 1.

In the pharmacophore-based QSAR models, the structural components of the ligands are represented by pharmacophore features with a specified radius. Only the pharmacophore features that are present in the hypothesis are used in the QSAR model. As for the atom-based models, the independent variables used in the QSAR model are the mN occupancies of the cubes by the m pharmacophore feature types: each variable corresponds to a given cube and a given feature type, and can take the value 0 or 1.

Once the occupancies are determined, a partial least-squares (PLS) regression analysis is applied to these binary-valued variables to obtain the QSAR model. Technical details on the regression analysis and the statistical measures used in the QSAR model are given in Appendix A.

Atom-based models are useful when features other than the pharmacophores are important to activity, such as steric clashes. However, their performance generally decreases as the diversity of the training set increases. If the structures in the training set contain a relatively small number of rotatable bonds and some common structural framework, then an atom-based model may work quite well.

Pharmacophore-based models assume that the activity is explained entirely by the pharmacophore model itself, and therefore cannot predict activities where other features are important to activity, such as steric clashes. If the structures in the training set are highly flexible or if they exhibit significant chemical diversity, a pharmacophore-based model may be more appropriate.

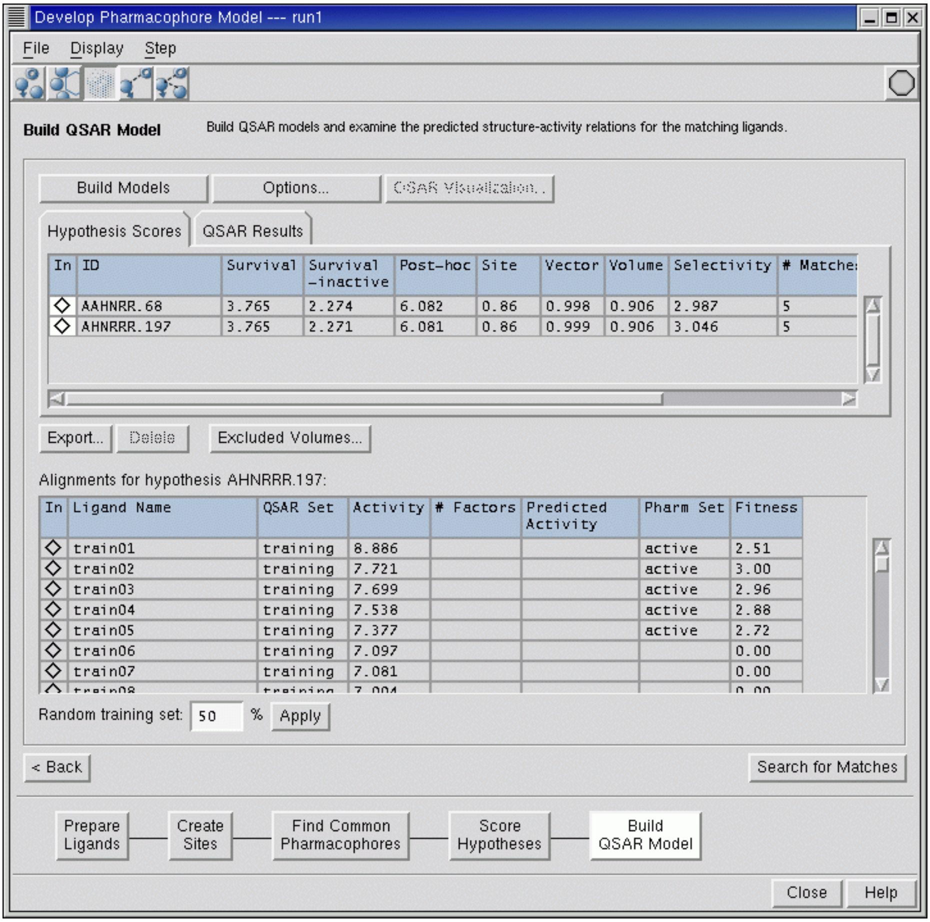


Figure 8.1. ***The initial view of the*** Build QSAR Model ***step.***

8.2 Choosing a Training Set and a Test Set

The first task in this step is to choose a training set and a test set, and exclude ligands that you do not want in either set. To display the ligands in the Alignments table, click the In column for any hypothesis in the Hypothesis Scores table. It does not matter which hypothesis you select, because all ligands are listed for all hypotheses. Initially, all ligands are included in the training set, and all rows are colored dark gray, which indicates that there is no corresponding QSAR model. The data columns are empty, and are filled in after the QSAR models are built.

To change the set membership of an individual ligand, click in the QSAR Set column for the ligand. The membership cycles between training, test, and blank, the last of which means that the ligand is excluded from both sets—that is, it is not used. To change the set membership for a group of ligands, select the ligands in the table using shift-click or control-click, then control-click in the QSAR Set column for any of the ligands.

You can select a random fraction of the ligands for the training set by entering a percentage in the Random training set text box and clicking Apply. The specified percentage of ligands is selected at random from the existing training and test sets and assigned to the training set. The remainder are assigned to the test set. Ligands that are in neither set are not used in the selection. The seed for the random selection can be set as an option—see the next section.

8.3 Specifying Options for the QSAR Model

If you want to select the model type, set the grid spacing, choose the maximum number of PLS factors, or specify a seed for random selection of the training set, you can do so in the Build QSAR Model - Options dialog box, which you open by clicking Options.

If you select the training set randomly, you may want to do this in a reproducible way. By default, the random seed changes each time a random training set is selected, so you get a different training set each time you click Apply in the Build QSAR Model step. However, if you change the value in the Random seed text box to any positive integer, you can ensure that the same random training set will be created each time you click Apply. The default value of zero ensures that the assignment is always random.

The QSAR model partitions the space occupied by the ligands into a cubic grid. Any structural component can occupy part of one or more cubes. A cube is occupied by an atom or a feature if its centroid is within the radius of the atom or feature. You can set the size of the cubes by changing the value in the Grid spacing text box. The allowed range is 0.5 Å to 2.0 Å.

The regression is done by constructing a series of models with an increasing number of PLS factors. The accuracy of the models increases with increasing number of PLS factors until over-fitting starts to occur. The maximum number of PLS factors is N/5, where N is the number of ligands. You can adjust this value in the Maximum PLS factors text box.

To select the type of model, choose Atom-based or Pharmacophore-based. In the pharmacophore-based model, the pharmacophore features are represented by spheres whose radii is given in the Tolerance column of the Feature radii table. The features are those defined in the Create Sites step. You can change the feature radii by editing the values in the Tolerance column.

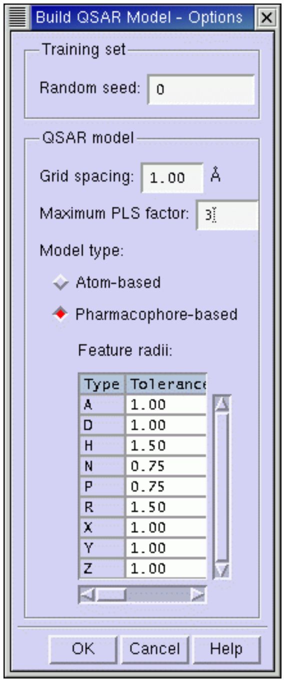


Figure 8.2. The Build QSAR Model - Options ***dialog box.***

8.4 QSAR Model Results

After you have selected the test set and the training set and set any options, click Build Models. A Start dialog box is displayed, in which you can adjust job settings. When you click Start, the job is run. The predicted activities are displayed in the Alignments table (see Table 8.1), and parameters for the quality of the fit are displayed in the QSAR results table (see Table 8.2). These parameters are defined in Section A.2 on page 166. Each row presents a regression model with a given number of PLS factors.

If you have more than one PLS factor in the model, you should examine the models produced to select the best model. For example, you can examine the predicted activities for the test set, and see at what point they begin to degrade, or you can compare the training set errors with the experimental uncertainty in the data.

The QSAR models are stored with the run, and can be used in the database search to predict activities for the hits. If you change the training set or the model parameters and build new QSAR models, these models overwrite the previous models. To save QSAR models for later use, you can export them with the hypothesis (click Export). The QSAR models are stored with the same file stem as the hypothesis data. If you intend to export more than one QSAR model

Table 8.1. Description of the Alignments table columns.

Column	Description
In	Inclusion status of the ligand. The diamond has a cross in it if the ligand is included in the Workspace, and is empty if the ligand is excluded. You can include and exclude ligands with click, shift-click, and control-click.
Ligand Name	The name of the ligand.
QSAR Set	Indicates whether a ligand is in the training set, the test set, or neither (the ligand is ignored). The column is blank if the ligand is ignored. Click the column repeatedly to cycle through the three possible states.
Activity	The ligand's activity. You can alter the activity values by directly editing the table cells.
# Factors	Number of PLS factors used for the QSAR model.
Pharm Set	Indicates the status of a ligand in the set used to build the pharmacophore model.
Predicted Activity	Activity predicted by the QSAR model. The number of rows in this column for each ligand is equal to the number of PLS factors specified in the Build QSAR Model - Options dialog box. Each row contains the prediction from a model containing the number of PLS factors indicated in the # Factors column.
Fitness	Fitness score from the scoring step.

Table 8.2. Description of the QSAR results table columns.

Column	Description
In	Inclusion status of the hypothesis. The diamond has a cross in it if the hypothesis is included in the Workspace, and is empty if the hypothesis is excluded. You can include and exclude hypotheses with click, shift-click, and control-click.
# Factors	Number of factors in the partial least squares regression model.
SD	Standard deviation of the regression.
R-squared	Value of R^2 for the regression.
F	Variance ratio. Large values of F indicate a more statistically significant regression
P	Significance level of variance ratio. Smaller values indicate a greater degree of confidence.
RMSE	Root-mean-square error.
Q-squared	Value of Q^2 for the predicted activities.
Pearson-R	Pearson R value for the correlation between the predicted and observed activity for the test set.

for a given hypothesis, you must provide a different name for each copy of the hypothesis, or store each in a different location.

You can also add excluded volumes to the hypothesis before exporting it. The Excluded Volumes button opens the same dialog box as in the previous step. See Section 7.6 on page 75 for more information.

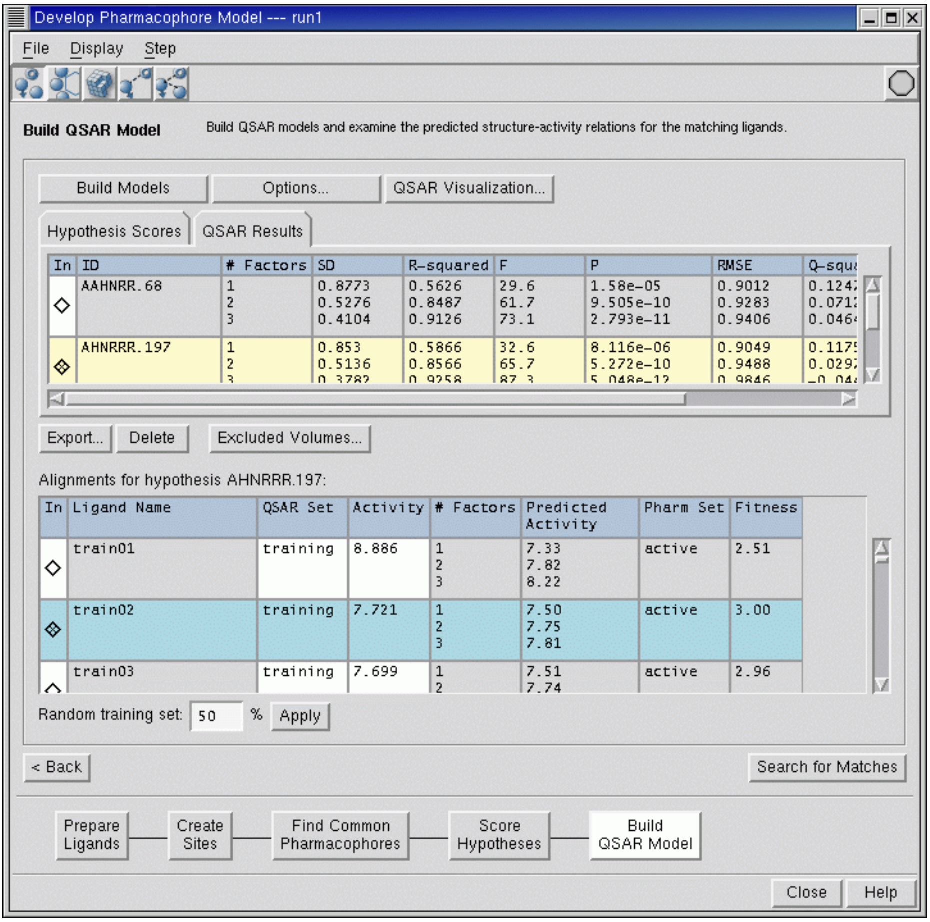
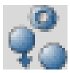


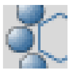
Figure 8.3. The Build QSAR Model ***step showing results of model-building.***

8.5 Viewing the QSAR Model

Once you have a QSAR model for a hypothesis, you can examine its 3D characteristics by displaying the QSAR model, the ligands, and the hypothesis in the Workspace. Each of these can be displayed independently. To display the hypothesis, the excluded volumes, or the QSAR model, click the appropriate toolbar button or choose the appropriate item from the Display menu. The buttons are described below:



View Hypothesis
Displays the selected hypothesis in the Workspace, as a spatial arrangement of feature symbols. For a description of these symbols, see Table 5.1 on page 51.



View Excluded Volumes
Displays excluded volumes for the selected hypothesis in the Workspace.

View QSAR Model
Displays the QSAR model for the selected hypothesis.

When you display the QSAR model, the cubes that represent the model are displayed in the Workspace, colored according to the sign of their coefficient values: blue for positive coefficients and red for negative coefficients. Positive coefficients indicate an increase in activity, negative coefficients a decrease. You can use the visualization of the coefficients to identify characteristics of ligand structures that tend to increase or to decrease activity.

In addition to viewing the model as a whole, you can examine the spatial distribution of contributions to the model by ligand, and by atom class or pharmacophore type. These capabilities are available in the QSAR Visualization Settings panel, which you open by clicking QSAR Visualization Settings. The visualization tools provided in this panel help you to identify features of ligand structures that are likely to contribute to higher or lower activity.

For example, if you select Workspace ligands under View volume occupied by and choose an atom type from the Selected atom class list, you can include the ligands in the Workspace one by one (click the In column of the Alignments table) and see which parts of all ligands have a positive or a negative contribution to the activity due to the chosen atom type. This might give a clue as to what functional groups are desirable or undesirable at certain positions in a molecule.

The QSAR Visualization Settings panel also has controls for the display of the model. For example, if you want to filter out cubes that have small coefficients, and therefore do not affect the activity much, you can use the sliders for the positive and negative coefficients in the Regression Coefficient Visualization section.

The three sections of the QSAR Visualization Settings panel are described below.

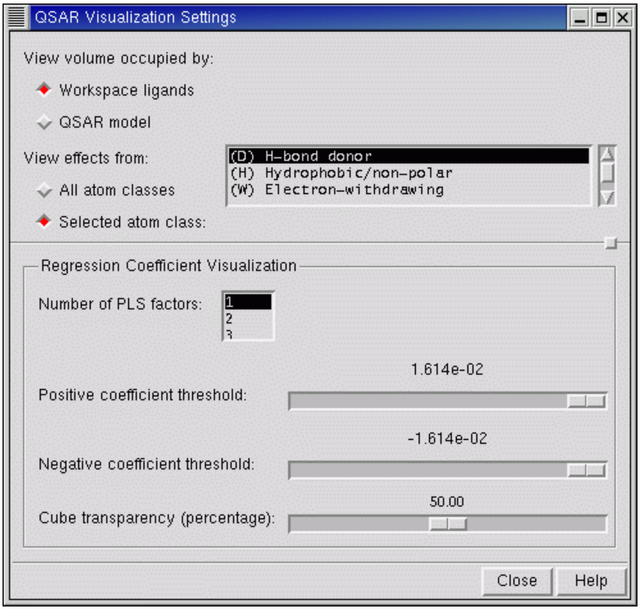


Figure 8.4. The QSAR Visualization Settings ***panel.***

View Volume Occupied By

The two options in this section allow you to choose whether to view the volume occupied by the QSAR model or by the ligands that are included in the Workspace.

- Workspace ligands— Display the cubes of the QSAR model grid that are occupied by the ligands that are in the Workspace.
- QSAR Model— Display all the cubes that are occupied in the QSAR model.

View Effects From

The options in this section allow you to choose whether to view the effects from all classes of atoms, or only a selected class of atom.

- All atom classes— Display the cubes of the QSAR model grid for all atom classes. The coefficients for each atom class are summed for the visualization.
- Selected atom class—Display the cubes of the QSAR model grid for the atom class that is selected in the list.

Regression Coefficient Visualization

This section provides controls for the choice of QSAR model and the display of its coefficients. For the coefficient sliders, cubes that have coefficients that are smaller in magnitude than the threshold are not displayed. This means that the coefficients that have the maximum magnitude are always displayed.

- Number of PLS factors—Select the number of PLS factors from the list to determine which QSAR model is displayed.
- Positive coefficient threshold—Adjust the threshold for the display of positive regression coefficients.
- Negative coefficient threshold—Adjust the threshold for the display of negative regression coefficients.
- Cube transparency—Adjust the transparency of the cubes from 0% (opaque) to 100% (transparent).

8.6 Continuing from the Build QSAR Model Step

When you have finished building a QSAR model, you can close the Build Pharmacophore Model panel, export the hypothesis with its QSAR model, continue directly to searching a database, or return to the previous step and select another set of hypotheses with which to build QSAR models.

To search a database, click Search for Matches. This button opens the Find Matches to Hypothesis panel, in which you can start a database search for ligands that match a hypothesis. All selected hypotheses are loaded by default, and the first of these is selected in the Find Matches to Hypothesis panel.

To build a QSAR model for another set of hypotheses, click Back, or click Score Hypotheses in the Guide. You can then select a hypothesis, and click Next or Build QSAR Model in the Guide to return to this step. When you do so, you are prompted to create a new run in which to store the hypotheses and the QSAR models. Each set of QSAR models is stored with its set of hypotheses in a separate run, so you can generate a QSAR model for as many hypotheses as you want. You can only store one QSAR model (with its PLS factors) for a given hypothesis in the run, but you can always export the hypothesis (click Export Hypothesis) with the current model if you want to store more than one QSAR model for a given hypothesis, or create a new run.

8.7 Step Summary

To build QSAR models:

1. Display the ligands in the Alignments table.
2. Select the training set and the test set.
3. (Optional) Choose a model and set parameters in the Build QSAR Model - Options dialog box.
4. Click Build Models.

To proceed to searching for matches:

1. Select the desired hypotheses in the Hypotheses table.
2. Click Search for Matches.

Building and Editing Hypotheses

In the Develop Pharmacophore Model workflow, hypotheses are generated from a set of active molecules, automatically taking into account common features and excluding features that are not common. The process does not directly take into account any explicit knowledge about the binding of a particular molecule to the receptor. (Of course, you can display the possible hypotheses and select the ones that fit with your understanding of the binding.)

Phase provides the means to use knowledge of ligand binding directly in the construction of a hypothesis from a single molecule (the reference ligand). For this molecule, Phase generates all possible pharmacophore sites from a set of pharmacophore features. You then select the sites that are included in the hypothesis. The features are the same as those used in the Develop Pharmacophore Model workflow, and can be supplemented with custom features or custom patterns in the same way. You can also edit hypotheses that were exported from the Develop Pharmacophore Model workflow.

You can add excluded volumes to the hypothesis that you construct. QSAR models, however, require a set of ligands with activities, which are not available in this workflow.

9.1 The Edit Hypotheses Panel

The Edit Hypotheses panel provides controls for creating, editing, deleting, importing, and exporting hypotheses based on an individual structure. To open the Edit Hypotheses panel, choose Edit Hypotheses from the Phase submenu of the Applications menu.

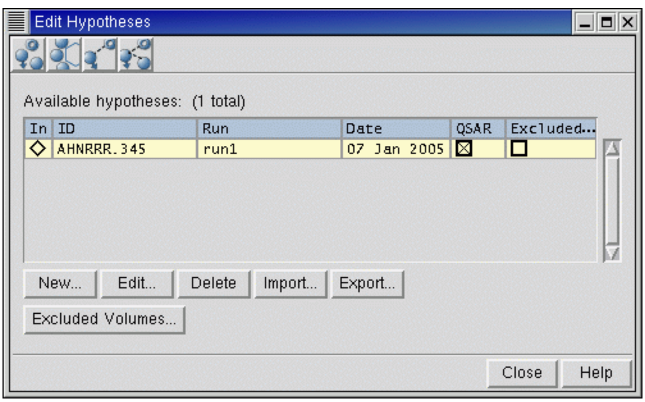
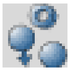
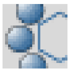


Figure 9.1. The Edit Hypotheses ***panel.***

The panel consists of a toolbar, a table of available hypotheses, and a set of action buttons. The toolbar contains four buttons, which are the same as in the Develop Pharmacophore Model panel, and allow you to view the hypothesis, the excluded volumes, and the intersite distances and angles:



View Hypothesis
View the selected hypothesis in the Workspace as a set of colored objects, which are described in Table 5.1 on page 51.



View Excluded Volumes
View excluded volumes for the hypothesis in the Workspace as a set of spheres.

View Distances
Display lines between site points and label them with intersite distances.

View Angles
Display lines between site points and label them with intersite angles.

The Available hypotheses table lists the hypotheses that are available for editing. This table operates on the same data used in the Find Matches To Hypothesis panel for searching a database; therefore any changes that you make here are also changed in the Find Matches To Hypothesis panel.

You can select a single row of the table, for editing, deleting, adding excluded volumes, or export. The columns of the table are described in Table 9.1. The table is noneditable.

Table 9.1. Description of the Available hypotheses table.

Column	Description
In	Inclusion status of the reference ligand and its hypothesis data. The diamond has a cross in it if the ligand is included in the Workspace, and is empty if the ligand is excluded. You can include and exclude ligands with click, shift-click and control-click.
ID	The identifier of the hypothesis. For new hypotheses, the identifier is constructed automatically from the feature letters and the entry name of the reference ligand. For hypotheses that were exported from a pharmacophore model development run, the identifier is the identifier from the run.
Run	The name of the run from which the hypothesis originated. For hypotheses created directly from a reference ligand, this is a series of dashes.
Date	The date that the hypothesis was last modified.
QSAR	Indicates whether a hypothesis has an associated QSAR model.
Excluded Volumes	Indicates whether a hypothesis has associated excluded volumes.

The action buttons, with the exception of the Delete button, open dialog boxes in which you can make the appropriate choices to perform the action. The buttons are described in Table 9.2.

Table 9.2. Action buttons in the Edit Hypotheses panel

Button	Action
New	Create a new hypothesis. Opens the Choose Reference Ligand dialog box, then the New Hypothesis dialog box.
Edit	Edit the selected hypothesis. Opens the Edit Hypothesis dialog box.
Delete	Delete the selected hypothesis from the table and from the project. Disk files for hypotheses that were imported are not removed, so you can always re-import them. Likewise, hypotheses are not removed from the project.
Import	Import a hypothesis from disk. Opens the Import Hypothesis dialog box, in which you can navigate to the desired hypothesis.
Export	Export the selected hypothesis to disk. Opens the Export Hypothesis dialog box, in which you can navigate to a location to save the hypothesis.
Excluded Volumes	Add excluded volumes to the selected hypothesis. Opens the Excluded Volumes dialog box. See Section 7.6 on page 75 for more information.

9.2 Creating New Hypotheses

New hypotheses are created using a set of pharmacophore features. Maestro uses the feature definitions to identify all the possible pharmacophore sites in the reference ligand. You can then choose which sites you want to include in the hypothesis. You can also edit the feature definitions using the Edit Features dialog box. See Section 5.2 on page 52 for more information on editing feature definitions.

To create a new hypothesis:

1. Click New.

   The Choose Reference Ligand dialog box is displayed. The table in the center of the dialog box lists entries from the Project Table. You can control which entries are displayed by choosing an item from the Choose entry from option menu. You can also sort the table by clicking one of the column headers or by clicking Sort by Project Table order.

2. Select an entry from the list.

   The entry name appears in the Name text box. You should ensure that the structure in the entry is a 3D, all-atom structure. If it is not, the pharmacophore features are likely to be incorrectly assigned.

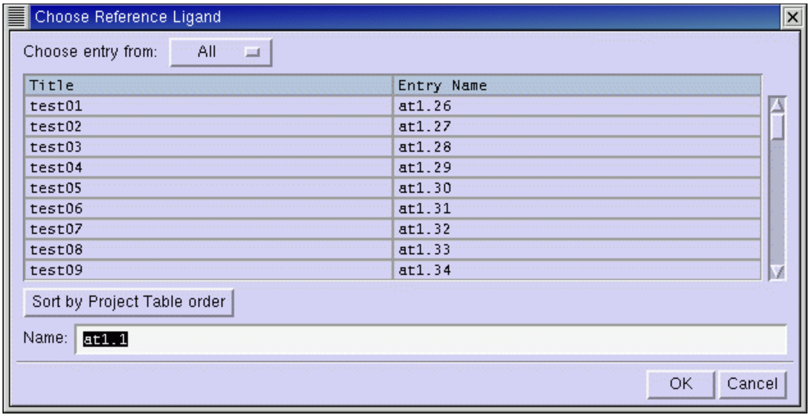


Figure 9.2. The Choose Reference Ligand ***dialog box.***

3. Click Choose.

   The Choose Reference Ligand dialog box closes and the New Hypothesis dialog box opens. This dialog box has two site lists: one of available sites in the reference ligand, and one of sites selected for the hypothesis, which is initially empty. You can choose which sites are displayed in the Workspace in the Mark sites section. By default, all sites are marked. This dialog box has the same controls as the Edit Hypothesis dialog box, shown in Figure 9.3.

4. Select the sites you want to include in your hypothesis from the Ligand sites list.

   You can select multiple sites by shift-clicking and control-clicking. Once you have selected sites, the Add button becomes available.

5. Click Add.

   The selected sites are added to the Hypothesis sites list. You can also select sites one by one and add them, and you can remove sites from the list. The Ligand sites list does not change when you add or remove sites.

6. When you have selected the desired sites, click OK.

   The New Hypothesis dialog box closes, and the hypothesis is displayed in the Workspace and added to the Available Hypotheses table of the Edit Hypotheses panel.

9.3 Editing Existing Hypotheses

As well as creating new hypotheses, you can edit hypotheses, including hypotheses that were exported from the Develop Pharmacophore Model panel. You cannot edit hypotheses from a pharmacophore model development run directly: you must export them first, then you can edit the exported version.

To edit an existing hypothesis:

1. Select the hypothesis in the Available hypotheses table, and click Edit.

   The Edit Hypothesis dialog box is displayed. This dialog box is identical to the New Hypothesis dialog box. You can choose which sites are displayed in the Workspace in the Mark sites section. By default, the hypothesis sites are marked.

2. Select the ligand sites you want to add to your hypothesis, and click Add.

   The selected sites are added to the Hypothesis sites list. You can select multiple sites by shift-clicking and control-clicking. The Add button is only available when you have selected one or more sites in the Ligand sites list.

3. Select the hypothesis sites you want to remove from your hypothesis, and click Remove.

   The selected sites are removed from the Hypothesis sites list. You can select multiple sites by shift-clicking and control-clicking. The Remove button is only available when you have selected one or more sites in the Hypothesis sites list.

4. When you have made the desired changes, click OK.

   The Edit Hypothesis dialog box closes, and the changes to the hypothesis are applied.

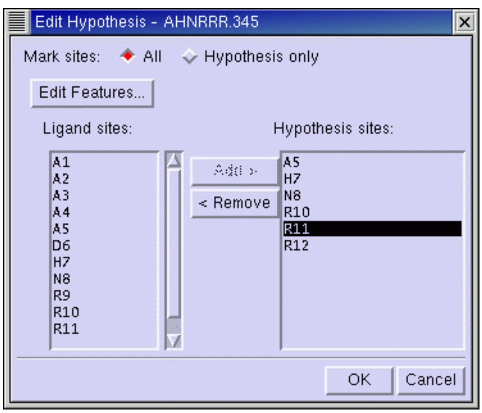


Figure 9.3. *The* Edit Hypothesis *dialog box.*

Preparing and Managing 3D Databases

To search for matches to a hypothesis, it is often convenient to store the structures you want to search through in a prepared 3D database. When you have prepared the database, the structures will be all-atom structures with reasonable 3D geometries. In addition, you can generate conformers and add site points for a given set of pharmacophore features to the structures. The database can be organized by adding subsets that define a range of structures for searching. Once you have prepared a database, you can add structures or remove structures, export structures to the Project Table, edit features, and recreate pharmacophore sites. These tasks can all be performed in the Manage 3D Database panel. 3D databases can also be managed from the command line—see Section 12.3 on page 129.

10.1 The Manage 3D Database Panel

The Manage 3D Database panel provides the tools for preparing and managing a structure database that can be searched for matches to a hypothesis. To open the Manage 3D Database panel, choose Manage 3D Database from the Phase submenu of the Applications menu in the main window.

When you open the Manage 3D Database panel for the first time, the New 3D Database dialog box is displayed. In this dialog box, you can create a new database by entering or browsing to a directory for the database and entering a database name. If you click Open Existing Database, a database selector is displayed, in which you can navigate to the desired database.

Once you have opened a database, the last database you used is stored in your Maestro preferences file, and this database is automatically opened when you open the panel. You can create a new database by choosing New from the Database menu. The New 3D Database dialog box is displayed when you do so; the Open Existing Database button is no longer in the dialog box. You can open an existing database by choosing Open from the File menu.

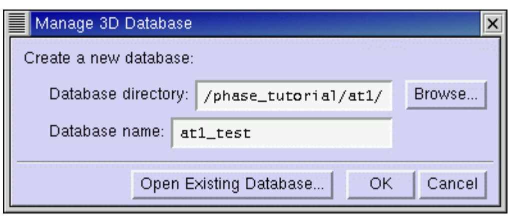


Figure 10.1. The New 3D Database ***dialog box.***

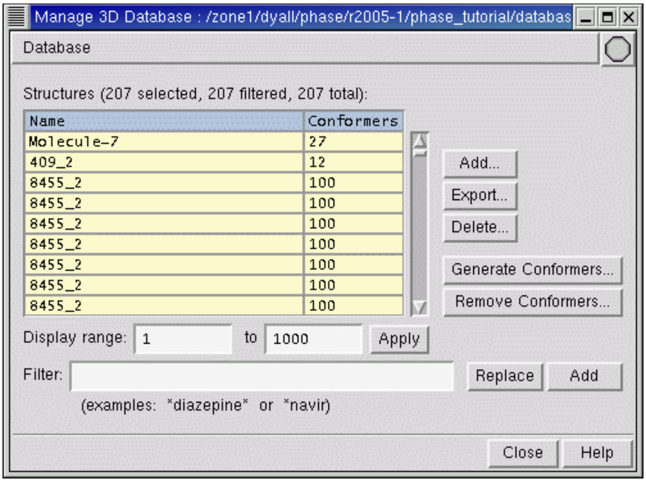


Figure 10.2. The Manage 3D Database ***panel.***

The Manage 3D Database panel has a menu bar, an octagon button, a table of structures, buttons for adding, deleting, and exporting structures; buttons for generating and removing conformations; and filtering controls. The Database menu items are described in Table 10.1. The octagon button turns green and spins when a job is running; clicking the button when it is active opens the Monitor panel.

Table 10.1. Database menu items

Item	Description
New	Opens the New 3D Database dialog box. In this dialog box, you can specify a database directory and a name for the database. The database is stored in the directory specified. You can navigate to the database directory by clicking the Browse button.
Open	Opens a file selector in which you can navigate to a directory and select it. The Selection field must contain a directory when you click Open. This directory must be the directory in which the database is stored.
Edit Features	Opens the Edit Features dialog box. In this dialog box you can import and edit feature definitions.
Create Sites	Opens the Create Sites dialog box, in which you choose whether to create pharmacophore sites for all structures or only those for which sites already exist. The sites are created when you click OK.
Subsets	Opens the Subsets dialog box, in which you can create subsets of structures for use in restricting the search for matches to a hypothesis.

10.2 Adding, Deleting, and Exporting Structures

Structures can be added to a database from an external file in Maestro or SD format, or from another Phase database. When you click Add, the Add Structures dialog box is displayed. In this dialog box you can either specify the file name of, or browse to, the file or the Phase database that contains the structures. You must also specify whether the file contains individual structures or one or more conformer sets.

When you add structures from another database, the feature definitions of the two databases are compared. If there is a mismatch, you can choose to use the definitions from the source database or from the destination database, or cancel the operation. For either choice of feature definitions, a job is run to create the sites for the structures that do not have the correct sites.

If you add structures from a file that contains sets of conformers, these are added as sets to the database and the sites are generated for each conformer. No further processing is performed, so you must ensure that the structures have been properly prepared and are in fact conformer sets, with the same atom order, connectivity, and title. Stereoisomers must differ at least in the title, in order to be detected as belonging to a different conformer set. Conformers must be consecutive structures in the file. You can generate conformer sets with a MacroModel conformational search, for example.

If the file contains individual structures, the Clean Structures dialog box is displayed when you click OK. A 3D database should be built from all-atom structures with chemically reasonable 3D geometries. The input structures could be represented in 2D form, without explicit hydrogen atoms, or with counter ions and solvent molecules. In addition, the structures might not have chirality information or be in the appropriate ionization state for physiological conditions. If any of these is the case for your input structures, you must run Clean Structures to obtain structures that are suitable for database searching. If the structures are already all-atom 3D structures, you can select Structures have already been cleaned. Otherwise you should set options for cleaning the structures and performing structural variations. Apart from the option at the top, this dialog box is identical to the Clean Structures dialog box opened from the Develop Pharmacophore Model panel. For details on the options, see Section 4.2 on page 41.

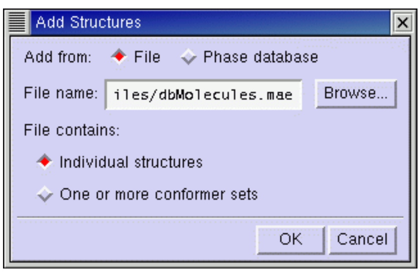


Figure 10.3. The Add Structures ***dialog box.***

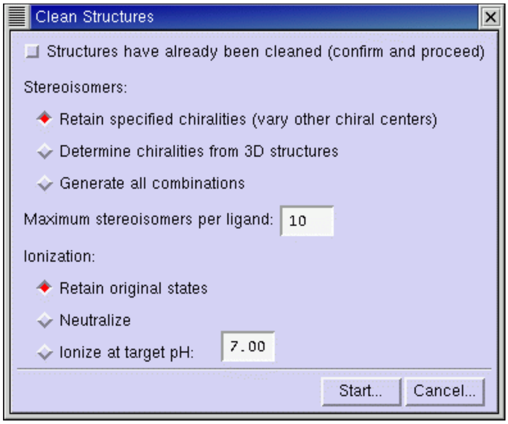


Figure 10.4. The Clean Structures ***dialog box.***

When you have selected the desired options, click Start. The Start dialog box is displayed, in which you can set job options and start the Clean Structures job. This job either checks for invalid structures, if the structures have already been cleaned, or performs the cleaning. When the job finishes, the structures are listed in the table in the Manage 3D Database panel.

To delete structures from the database, select the structures in the table and click Delete. A dialog box is displayed so that you can confirm or cancel the deletion.

To export structures to the Project Table, select them in the table and click Export. You can add the entire conformer set for each structure, or only the lowest-energy conformation.

Note that you can only select structures that are visible in the table. If the structures you want are not visible, use the Filter and Display Range controls to list the desired structures. See Section 10.6 on page 102 for more information.

10.3 Generating Conformers and Sites

Phase provides some flexibility in the generation of conformers and creation of site points for the structures in the database. You can generate the conformers and the sites (site points) during database preparation and store them with the database, or you can generate them as needed during database searching. A database can contain both single structures, without conformers and sites, or structures that have conformers and sites.

Storing the conformers and sites as a part of the database increases the disk space needed, but it makes the search faster. When you generate the conformers and site points as part of database preparation, you can perform a more thorough conformational search including energy minimization of the conformers using the interface to MacroModel. This option is not available when they are generated as needed. If you intend to use the same set of feature definitions for a variety of database searches, you should consider storing the sites as part of the database.

Generating conformers and sites during database searching uses much less disk space, but it makes the search slower, and does not allow postminimization of conformers with MacroModel. This option is useful if you do not have sufficient disk space for the full database, or if you plan to use different feature sets for database searches. However, if you have stored a database with site points for a given feature set, you can always use a different feature set and generate the site points during the search. If you do so, Phase allows you to discard the site points after the search, or store them, replacing the original site points in the database.

To generate conformers, click Generate. The Generate Conformations dialog box is displayed. This dialog box is the same as the dialog box opened from the Develop Pharmacophore Model panel, with the exceptions that the Current conformations options are absent and the only search method available is Ligand Torsional Search. If you want to generate conformers with a different conformational search method, you can run the conformational search independently (for example, with MacroModel) and import the conformers as a set. For details on using the Generate Conformers dialog box, see Section 4.3 on page 43.

After setting the desired options, click Start. A dialog box is displayed in which you can choose the host to run the job. You can distribute this job over multiple processors. While it is running, the octagon in the top corner of the Manage 3D Database panel turns green and spins. When the job finishes, a second job is run to generate the site points for each conformer of each structure. When the site generation job finishes, the octagon stops and turns gray and the number of conformers is displayed in the Conformers column of the Ligands table.

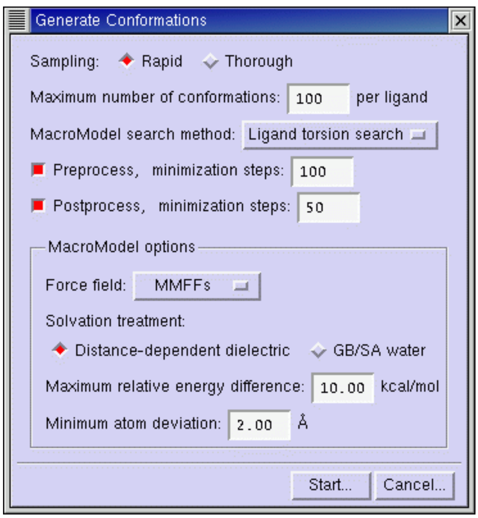


Figure 10.5. The Generate Conformations ***dialog box.***

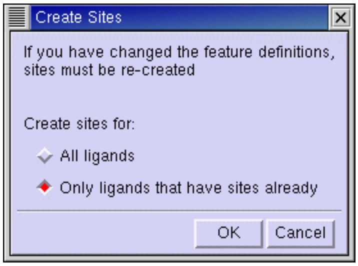


Figure 10.6. The Create Sites ***dialog box.***

Sites are automatically created when conformers are generated or conformer sets are imported. Normally, you will not need to explicitly create sites. If you change the feature definitions you should recreate the sites. To do so, choose Create Sites from the Database menu, and in the Create Sites dialog box (Figure 10.6), choose whether to create sites for all structures or only for those that already have sites.

If you want to delete the conformers of several structures and keep only the lowest-energy conformer, select the structures in the table and click Remove. A dialog box is displayed so that you can confirm or cancel the deletion.

10.4 Selecting and Editing Features

If you do not want to use the standard feature set, you can edit the features or import features from another location. To do so, choose Edit Features from the Database menu. The Edit Features dialog box is displayed, in which you can change the feature definitions or import features. This dialog box is identical to the dialog box opened from the Develop Pharmacophore Model panel. See Section 5.2 on page 52 for information on using this dialog box.

When you click OK in the Edit Features dialog box, the Create Sites dialog box is displayed. If you click Cancel in the Create Sites dialog box, a warning is displayed, and the Edit Features dialog box is redisplayed. You must either create sites or discard the new or changed features.

10.5 Creating Subsets

When you search the 3D database for matches to a pharmacophore model, it can be useful to restrict the search to a subset of structures. You can create subsets in the Subset panel, which you open by choosing Subset from the Database menu. Subsets are lists of structures, and can be used to restrict the database search. The available subsets are listed in the table in the Subset panel.

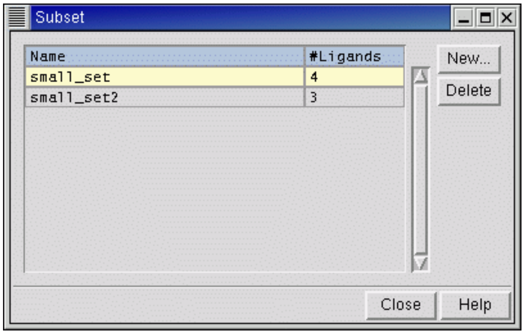


Figure 10.7. The* Subset *panel.

To create a new subset, click New. In the New Subset dialog box, enter the name of the subset, and select a source for the subset. There are three options for the source:

- Currently selected ligands—Use the ligands that are selected in the Manage 3D Database panel.
- Text file—Use the ligands whose names are listed, one per line, in the specified text file. You can enter the file name in the File text box or click Browse to navigate to the file.
- Results of a prior search analysis—Use the hits from a previous search of the current database. The names are taken from the Maestro file that contains the hits. You can enter the file name in the File text box or click Browse to navigate to the file.

When you search the database, you can select one of these subsets by name to restrict the search to the subset.

To change the name of a subset, you can edit it directly in the table cell, but you must make it unique. Spaces and some other characters are removed from subset names. To delete the selected subset, click Delete.

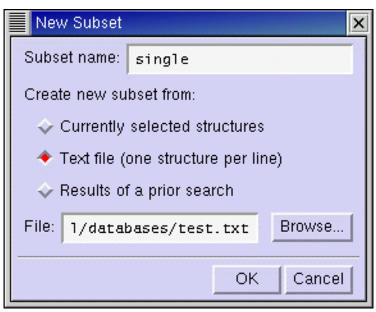


Figure 10.8. The* New Subset *dialog box.

10.6 Filtering the Listed Structures

You can filter the listed structures by entering a text string in the Filter text box. The string can contain asterisks as wild-card characters, which are interpreted as zero or more characters. After entering text in the text box, click Replace to replace the structures in the table with those that match the filter, or click Add to add the structures that match the filter to the table. To redisplay all structures, enter "*" in the text box and click Replace.

In addition to the text filter, the list is limited to a numerical range of structures, which you can enter in the Display range text boxes. When you click Apply, structures in the range you specified are listed in the Structures table. The range is applied to the results returned by the filter. Once you apply another filter, the range is inactive until you click Apply again.

For example, suppose the structures are labeled `structure1` through `structure100`, and you applied the filter `*5*`, which returns every structure that includes a “5”. Limiting the range to 1 through 5 would display the first five filtered structures, which are the structures labeled `structure5`, `structure15`, `structure25`, `structure35`, and `structure45`.

Finding Matches to Hypotheses

When you have a pharmacophore model and a prepared 3D database or file of 3D structures, you can proceed to searching a database for structures that match the hypotheses of the model. The database search process is normally performed in two steps: *finding* and *fetching*.

In the find step, the database is searched for geometric arrangements of pharmacophore sites that match the site types and intersite distances of the chosen hypothesis. For example, the hypothesis DHRR contains one donor (D), one hydrophobe (H) and two aromatic rings (R1, R2). These four pharmacophore features give rise to six unique intersite distances: dDH, dDR1, dDR2, dHR1, dHR2 and dR1R2. The find step scans the database for occurrences of the four feature types for which the six intersite distances are sufficiently close to those of the hypothesis. When such an occurrence is found, information about the match is written to a *match file*.

In the fetch step, the match file is used as a lookup table to rapidly retrieve the relevant conformers from the database and align them to the hypothesis. We refer to these conformers as *hits*. When hits are fetched, they are ordered and filtered, so that only a fraction of the total number of matches is presented. The hits are ordered first by their fitness score, then filtered by number, and by occupation of excluded volumes. Finally, the activity is predicted if there is a QSAR model available. The hits that are fetched are added to the Project Table.

The find step is the center of the search, and is the most time-consuming part of the process. Phase separates the two steps so that you do not need to repeat the find step if you only want to apply different filters or a different ordering to the matches before they are retrieved.

In the Find Matches to Hypotheses panel you can search a database or a file for matches to a selected hypothesis. The panel provides options for matching and for processing the hits.

11.1 The Fitness Score

Hits are first fetched in order of decreasing *fitness*. Fitness is a score that measures how well the matching pharmacophore site points align to those of the hypothesis, how well the matching vector features (acceptors, donors, aromatic rings) overlay those of the hypothesis, and how well the matching conformation superimposes, in an overall sense, with the reference ligand conformation. The fitness score is defined by

$$S = W_{\text{site}}(1 - S_{\text{align}}/C_{\text{align}}) + W_{\text{vec}}S_{\text{vec}} + W_{\text{vol}}S_{\text{vol}} .$$

Table 11.1. Description of parameters in the fitness scoring function.

Parameter	Description
S_{align}	Alignment score: RMS deviation between the site point positions in the matching conformation and the site point positions in the hypothesis.
C_{align}	Alignment cutoff. User-adjustable parameter; default is 1.2.
W_{site}	Weight of site score. User-adjustable parameter; default is 1.0.
S_{vec}	Vector score: average cosine between vector features in the matching conformation and the vector features in the reference conformation.
W_{vec}	Weight of vector score. User-adjustable parameter; default is 1.0
S_{vol}	Volume score: Ratio of the common volume occupied by the matching conformer and the reference conformer, to the total volume (the volume occupied by both). Volumes are computed using van der Waals models of all non-hydrogen atoms.
W_{vol}	Weight of volume score. User-adjustable parameter; default is 1.0

The terms in the score are described in Table 11.1. This score is a truncated version of the survival score for hypotheses. See Section 7.1 on page 66 for more information on the various terms in the scoring function and how they are defined.

By adjusting the parameters in the fitness function, you can control the order in which hits are returned. For example, if you want to emphasize the alignment of vector features, you could increase the vector weight. If the overall molecular superimposition is most important, you could increase the volume weight.

11.2 Setting up A Search

Searches are performed from the Find Matches to Hypothesis panel. To open this panel, choose Find Matches to Hypothesis from the Phase submenu of the Applications menu. If you are working in the Score Hypotheses step or the Build QSAR Model step of the Build Pharmacophore Model panel, click Search for Matches.

To perform a search, you must select the source of the structures to search and a hypothesis, set any options for matching, and for filtering and treatment of the hits, then click Start. The options are described below.

Selecting a Structure Source

The collection of structures that you use to search for matches can come from one of three sources: a prepared 3D database, an external file, or the Project Table. The structures must be all-atom 3D structures.

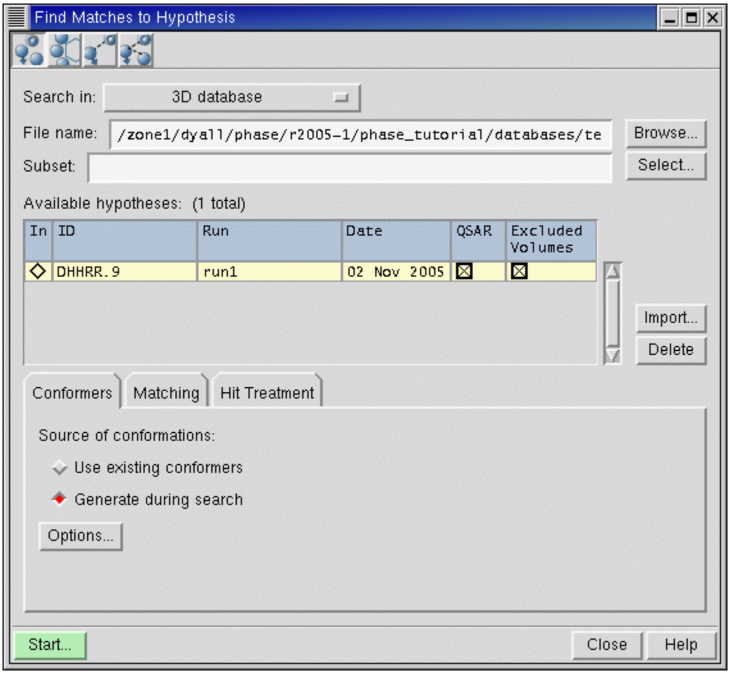


Figure 11.1. The Find Matches to Hypotheses ***panel.***

- To search a 3D database, choose 3D database from the Search in option menu, then either enter a file name in the File name text box, or click Browse and navigate to the desired database. If you want to search a subset of structures from the database, enter the name of the subset in the Subset text box, or click Select, and select a subset in the Select Subset dialog box. Subsets can be generated in the 3D database management process. By default, the database selected is the last database you used.
- To search the structures in a file, choose External file from the Search in option menu, then either enter a file name in the File name text box, or click Browse and navigate to the desired database. The file must be in Maestro or SD format.
- To search structures from the Project Table, select the entries you want to search in the Project table, then choose Project Table (selected entries) from the Search in option menu.

Selecting a Hypothesis

The hypotheses that are listed in the Available hypotheses table include hypotheses that were carried forward to the Build QSAR Model step in the Develop Pharmacophore Model panel, hypotheses created or edited in the Edit Hypotheses panel, and hypotheses that you import.

This table is linked to the table in the Edit Hypotheses panel: any changes you make here are reflected in the Edit Hypotheses panel, and vice versa. The columns are described in Table 9.1

To choose a hypothesis for the search, select the table row. You can delete a hypothesis from the Available hypotheses table by selecting it and clicking Delete. Deleting hypotheses here has no effect on the hypotheses in the run, only on what is available for searching.

You can add hypotheses to the table by importing them: click Import, and navigate to the desired hypothesis in the file chooser that is displayed. However, you can only import hypotheses that were previously exported. You cannot import hypotheses from another project.

You can display the hypothesis, its excluded volumes, and its intersite distances and angles in the Workspace by clicking the toolbar buttons. These buttons are the same as in the Develop Pharmacophore Model panel and the Edit Hypotheses panel, and are described in Section 9.1 on page 89.

The feature definitions for the hypothesis should match those in the database. If they do not match, you can compute pharmacophore sites as needed using the hypothesis feature definitions, without replacing the existing sites in a database.

If you open this panel from the Build Pharmacophore Model panel, the hypothesis that you were working with is selected in the Available hypotheses table.

Selecting the Source of Conformations

The search for matches to a hypothesis requires conformations of each molecule searched. If the source of structures includes sets of conformers, you can select Use existing conformers in the Conformers tab. If the source does not include conformers, select Generate as needed in the Conformers tab, to generate them during the search. The conformer generation uses the Ligand Torsion Search method (see Section 4.3 on page 43 for details), with a restricted set of options. You can set the options in the Generate Conformers dialog box by clicking Options.

Setting Search Criteria

The search mode and search criteria are set in the Matching tab.

- If you are starting with a new structure source, select Find new matches as the search mode.
- If you want to apply a different filter to an existing set of matches, or use a different scoring function, select Use existing matches.

When you search for matches to the hypothesis, you can set the number of site points that must match and select individual features that must match. The set of features that are required to match is known as a *site mask*.You can set tolerances for matching intersite distances and for matching features. Any intersite distance less than the specified value is considered to match.

The feature-matching tolerances can be set in the Feature Tolerances dialog box, which you open by clicking Feature Tolerances. Features are considered to match if they are within the specified distance of the features of the hypothesis. The values that you select can be saved as the default values. All these options are only available if you select Find new matches.

Setting Filtering and Scoring Options

At the end of the search, the matches are stored. They are then filtered to generate a reduced list of hits, and properties are calculated for these hits. Filters are set in the Hit Treatment tab.

- To calculate activities for the hits based on the QSAR model, if one is available, select Apply QSAR model. The hits are not ordered by the activities before they are returned, but you can sort by activity in the Project Table.
- To filter out matches that occupy excluded volumes, select Apply excluded volumes.
- To limit the number of hits returned, enter values in the Return at most text boxes. Some molecules can return more than one hit because different alignments or different conformers might match the hypothesis.
- To change the fitness scoring function, enter new values of the weights in the text boxes.

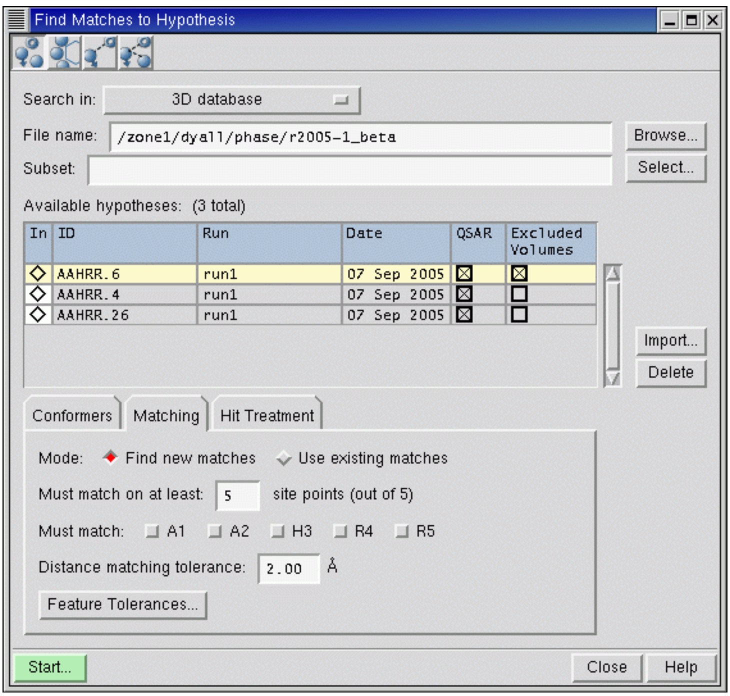
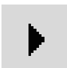


Figure 11.2. The Matching ***tab of the*** Find Matches to Hypothesis ***panel.***

11.3 Search Results

Each time you search the database, the hits are added as an entry group to the Project Table, where you can use the full range of applications and facilities available from Maestro. The fitness score and the activity predicted by the QSAR model (if any) are added as properties, along with a property that indicates which hypothesis was matched.

You can view the hits superimposed on the hypothesis by including them in the Workspace, and displaying the hypothesis from the Find Matches to Hypothesis panel, using the toolbar buttons. If you want to cycle through the hits, you can use the ePlayer: select the hits in the Project Table, then click the Play forward button:



You can change the speed at which the hits are displayed in the ePlayer Options dialog box, which you open from the ePlayer menu.

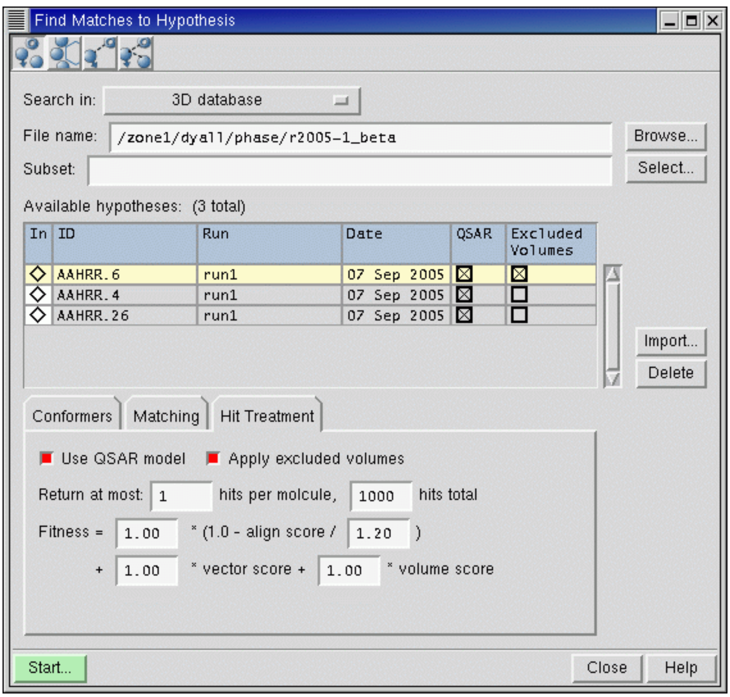


Figure 11.3. The Hit Treatment ***tab of the*** Find Matches to Hypothesis ***panel.***

Pharmacophore Model Development from the Command Line

To develop a pharmacophore model from a set of ligands, you must prepare the ligands, generate pharmacophore sites for the ligands, find common pharmacophores, then score the resultant hypotheses. You can also build QSAR models for any of the hypotheses. These five steps are referred to by the names used in the Phase panel in Maestro: Prepare Ligands, Create Sites, Find Common Pharmacophores, Score Hypotheses, and Build QSAR Model.

The first step, Prepare Ligands, is the only step that does not have Phase utilities to perform the task. The ligands you provide must be properly prepared and stored in one or more Maestro files prior to starting the workflow. Each molecule should be represented by multiple low-energy 3D structures that provide good coverage of that molecule's conformational space. See Chapter 10 and Chapter 11 of the *MacroModel User Manual* for information on creating conformational models, and see the *LigPrep User Manual* for information on 2D-to-3D conversion and structure variation. If you want to develop QSAR models, then each molecule that will be used to train models should contain an activity property, expressed either in concentration units or as –log[concentration].

Within each Maestro file, conformers for a single molecule must be stored consecutively. If two consecutive structures differ only in their stereochemistry, they are treated as conformers of a single molecule unless the titles for those two structures are different.

12.1 Workflow Summary

The complete command-line pharmacophore model development workflow is outlined below, in terms of the scripts to run. These scripts are described in detail in the next section; links to the relevant script are provided in the summary below. The starting point is one or more Maestro files containing multiconformer models for the ligands of interest. A Phase project is created from these ligands, after which a series of steps is followed, directly analogous to the Develop Pharmacophore Model workflow in Maestro.

Each step requires a Phase main input file, and other files that are stored in the current working directory (where you start the jobs), a subdirectory of this directory, or the Phase distribution. Most of the required job setup, including creation of the input files, is handled with the `-setup` options of the utilities listed, and cleanup of temporary and intermediate files is done with the `-cleanup` option. For details on the Phase main input file, see Section B.2 on page 172.

Output files are created in the current working directory, or specified subdirectories of this directory. Ligand structures and generated ligand-related information is stored in a subdirectory whose default name is `ligands`; this subdirectory is referred to as the *ligands directory*. Output of the scoring step is stored in a subdirectory whose default name is `result`; this subdirectory is referred to as the *results directory*.

The top-level programs that perform the work accept the standard Job Control options listed below. These are represented by *job-options* in the syntax statements.

`-HOST` *host*	Run the job on the specified host. Multiple hosts and processors can be specified with `-HOST "`*host1*`:`*nproc1* *host2*`:`*nproc2* ... `"`
`-LOCAL`	Run the job in the current directory, rather than in a temporary scratch directory.
`-TMPDIR` *tmpdir*	Use *tmpdir* for temporary files.
`-WAIT`	Do not return control to the shell until the job finishes.
`-INTERVAL` *N*	Interval in seconds between output updates.
`-NICE`	Run the job at reduced priority.

Create/Add to a Project:

```
$SCHRODINGER/utilities/pharm_project {-new|-add} [options]
```

Modify Master Data:

```
$SCHRODINGER/utilities/pharm_data [options]
```

Create Pharmacophore Sites:

```
$SCHRODINGER/utilities/pharm_create_sites -setup [setup-options]
$SCHRODINGER/phase_feature create_sites [job-options]
$SCHRODINGER/utilities/pharm_create_sites -cleanup
```

Find Common Pharmacophores:

```
$SCHRODINGER/utilities/pharm_find_common -setup [setup-options]
$SCHRODINGER/phase_partition find_common [job-options]
$SCHRODINGER/utilities/pharm_find_common -cleanup
```

Score Hypotheses with Respect to Actives:

```
$SCHRODINGER/utilities/pharm_score_actives -setup [setup-options]
$SCHRODINGER/phase_scoring score_actives [job-options]
$SCHRODINGER/utilities/pharm_score_actives -cleanup
```

Score Hypotheses with Respect to Inactives:

```
$SCHRODINGER/utilities/pharm_score_inactives -setup [setup-options]
$SCHRODINGER/phase_inactive score_inactives [job-options]
$SCHRODINGER/utilities/pharm_score_inactives -cleanup
```

Cluster Hypotheses by Geometric Similarity:

```
$SCHRODINGER/utilities/pharm_cluster_hypotheses -setup [setup-options]
$SCHRODINGER/phase_hypoCluster cluster_hypotheses [job-options]
$SCHRODINGER/utilities/pharm_cluster_hypotheses
    -cleanup [cleanup-options]
```

Build QSAR Models:

```
$SCHRODINGER/utilities/pharm_build_qsar -setup [setup-options]
$SCHRODINGER/phase_multiQsar build_qsar [job-options]
$SCHRODINGER/utilities/pharm_build_qsar -cleanup
```

Preserve Project Data in a Tar Archive:

```
$SCHRODINGER/utilities/pharm_archive [options]
```

Once pharmacophore hypotheses and QSAR models have been developed, a number of other command line utilities may be run:

Align Project Ligands or New Molecules to a Pharmacophore Hypothesis:

```
$SCHRODINGER/utilities/pharm_align_mol -setup [setup-options]
    -job jobname
$SCHRODINGER/phase_fileSearch jobname [job-options]
$SCHRODINGER/utilities/pharm_align_mol -cleanup -job jobname
```

Align or Merge a Pair of Hypotheses:

```
$SCHRODINGER/utilities/align_hypoPair [options]
```

Create Excluded Volumes Automatically:

```
$SCHRODINGER/utilities/create_xvolShell [options]
$SCHRODINGER/utilities/create_xvolClash [options]
$SCHRODINGER/utilities/create_xvolReceptor [options]
```

Analyze QSAR Predictions within Hit Files:

```
$SCHRODINGER/utilities/phase_qsar_stats [options]
```

Visualize QSAR Models:

```
$SCHRODINGER/utilities/qsarVis [options]
```

12.2 Pharmacophore Model Development Utilities

The pharmacophore model development utilities are stored in `$SCHRODINGER/utilities`. Except where noted, all changes to project files should be done only through the use of the utilities listed below. Brief descriptions of the use of the utilities is given below; full descriptions are given in the following sections.

`pharm_help`	Prints a help message summarizing the command line pharmacophore model workflow, including all the utilities that follow.
`pharm_project`	Creates a new command line pharmacophore model project and adds molecules to an existing project.
`pharm_data`	Performs various operations on the project data.
`pharm_create_sites`	Does setup/cleanup for the job that creates pharmacophore sites.
`pharm_find_common`	Does setup/cleanup for the job that identifies common pharmacophores.
`pharm_score_actives`	Does setup/cleanup for the job that scores hypotheses with respect to actives.
`pharm_score_inactives`	Does setup/cleanup for the job that scores hypotheses with respect to inactives.
`pharm_cluster_hypotheses`	Does setup/cleanup for the job that clusters hypotheses by geometric similarity.
`pharm_build_qsar`	Does setup/cleanup for the job that builds QSAR models.
`pharm_archive`	Preserves project data in a tar archive.
`pharm_align_mol`	Does setup/cleanup for the job that aligns project ligands or new molecules to a hypothesis.
`align_hypoPair`	Aligns/merges a pair of hypotheses.
`create_xvolShell`	Creates a shell of excluded volume spheres around one or more ligands. Provides a means of defining shape-based queries for database searching.
`create_xvolClash`	Creates excluded volumes using actives and inactives that have been aligned to a hypothesis. Excluded volumes are placed in locations that would cause steric clashes only for the inactives.
`create_xvolReceptor`	Creates excluded volumes using a receptor structure or a portion thereof.

`phase_qsar_stats`	Extracts statistics from a hit file that contains QSAR predictions.
`qsarVis`	Standalone graphical interface for visualizing QSAR models. Available only on Linux-x86 systems.

In addition to the above utilities, the following programs are in the `$SCHRODINGER` directory:

`phase_feature`	Creates pharmacophore sites.
`phase_partition`	Identifies common pharmacophores.
`phase_scoring`	Scores hypotheses with respect to actives.
`phase_inactive`	Scores hypotheses with respect to inactives.
`phase_hypoCluster`	Clusters hypotheses by geometric similarity.
`phase_multiQsar`	Builds 3D QSAR models for a collection of hypotheses, and generates a statistical summary for each model.
`phase_qsar`	Builds a single 3D QSAR model, and generates detailed output.
`phase_fileSearch`	Aligns structures in a single file to a hypothesis.

These programs accept the standard job options that are listed in Section 12.1.

12.3 Setting Up a Phase Pharmacophore Model Project

Phase pharmacophore model projects are a collection of files, managed by a utility called `pharm_project`. These projects are *not* the same as the corresponding Maestro projects, but the results of pharmacophore model development—the hypotheses—can be imported into Maestro. In addition to managing the structures in the project with the utility `pharm_project`, you can add or change certain data associated with the structures with the utility `pharm_data`. These two utilities are described in the next two sections.

12.3.1 pharm_project

Creates a new command line pharmacophore model project or adds ligands to an existing project. Conformations must be generated ahead of time and stored in a Maestro file. Consecutive structures with identical titles and connectivities will be treated as conformations of a single molecule.

Syntax

```
pharm_project {-new|-add} -mae maefile [-ignoreTitles] [-act actProp]
    [-conf confProp]
```

Options

`-new`	Create a new project in the current directory. Any existing project data will be removed upon confirmation.
`-add`	Add ligands to an existing project. All project data from Create Sites forward will be removed upon confirmation.
`-mae` *maefile*	Maestro file containing ligand conformations.
`-ignoreTitles`	Ignore titles when perceiving conformations. Consecutive structures with identical connectivities are treated as conformations of a single ligand, even if their titles differ.
`-act` *actProp*	The name of the activity property exactly as it appears in *maefile*. You must supply this information if you intend to use the ligands to build QSAR models.
`-conf` *confProp*	The name of the relative conformational energy property exactly as it appears in *maefile*. You must supply this information if you intend to score hypotheses with respect to relative conformational energy.

Output Files

`ligands/`	Subdirectory that holds all structural data for the project ligands.
`ligands/*.mae`	Individual ligand files split out from the input files.
`MasterData.tab`	A specially formatted text file that holds project data required in various steps of the workflow. Certain modifications are permitted (by hand or through the use of `pharm_data`).
`MasterData.backup`	A backup copy of `MasterData.tab`. Used to revert changes you make to `MasterData.tab`. Do not modify.
`ProjectLigands.inp`	Ligand records file. Provides a compact summary of project data, and serves as a template for creating subsets of ligands to align to a hypothesis. While the file can be modified without affecting the integrity of project, it is recommended that you leave it as is, and make a copy of the file if you need to define a subset.
`FeatureFreq.tab`	Feature frequency file. Sets minimum and maximum allowed feature frequencies for common pharmacophore perception.
`FeatureTol.tab`	Feature matching tolerances that can be applied when hypotheses are scored with respect to actives.
`pharma_feature.ini`	Default pharmacophore feature definitions. You may replace this file with customized definitions, but it is strongly recommended that you do the customization with the Phase interface in Maestro.

12.3.2 pharm_data

Performs various operations on `MasterData.tab` and propagates any changes in this file to the rest of the project. This includes changes that may have been made by hand.

Syntax

```
pharm_data [-log|-exp] [-multiply scale]
    [-active aboveVal] [-inactive belowVal]
    [-train numTrain [-rand seed [-pharm_set]]]
    [-conf confProp] [-commit] [-restore]
```

Options

`-log`	Perform $-\log_{10}$(ACTIVITY) conversion on the activity property.
`-exp`	Perform $10^{-\mathrm{ACTIVITY}}$ conversion on the activity property.
`-multiply` *scale*	Scale the activity property by *scale*. Done prior to the $-\log_{10}$(ACTIVITY) conversion and after the $10^{-\mathrm{ACTIVITY}}$ conversion.
`-active` *aboveVal*	Use the specified activity threshold to distribute ligands between `active` and `none` `PHARM_SET` categories. All ligands are affected.
`-inactive` *belowVal*	Use the specified activity threshold to assign ligands to the `inactive` `PHARM_SET` category. Only ligands with activities below the threshold are affected. May be used with the `-active` option.
`-train` *numTrain*	Distribute ligands between the training set (`train`) and test set (`test`) `QSAR_SET` categories. Only ligands with numeric activities are affected. By default, the first *numTrain* qualifying ligands are assigned to the `train` category.
`-rand` *seed*	Assign ligands to `train` and `test` `QSAR_SET` categories randomly using the supplied random seed integer. Valid only when `-train` *numTrain* is used.
`-pharm_set`	Consider the `PHARM_SET` membership when assigning random `QSAR_SET` categories. If this option is used, ligands for which `PHARM_SET` is `active` or `inactive` are always assigned to the `train` `QSAR_SET`, provided they have numeric activities. Valid only with `-rand` *seed*.
`-conf` *confProp*	The name of the relative conformational energy property exactly as it appears in the Maestro files supplied to `pharm_project`. Use this option if you want to score hypotheses with respect to relative conformational energy, but you did not define the property when `pharm_project` was run.

-commit	Commit changes in MasterData.tab to project, including ligand Maestro files. Any forward project data affected by these changes will be removed upon confirmation by the user.
-restore	Restore previous MasterData.tab file. You would normally do this when you decide that you do not want to remove forward data, such as when a -commit operation is aborted. May not be used in combination with any other option.

If you have completed any forward steps in the project workflow, the results generated in those steps may be invalidated by changes you make to MasterData.tab. When you attempt to commit the changes, you will be supplied with a list of files from forward steps that will be invalidated, and you will be given a chance to abort the commit operation. If you choose to abort, you can rerun pharm_data with the -restore flag to revert to the previous version of MasterData.tab (i.e., the data stored in MasterData.backup).

If your activities are expressed in concentration units (e.g. K_i or IC_{50} values) and you intend to create QSAR models, then you must use the -log and -commit options. You must also perform the -log conversion on concentrations if you plan to assign PHARM_SET categories using the -active or -inactive options, because these assignments are based on the assumption that the ACTIVITY property increases as potency increases.

12.4 Creating Sites

Pharmacophore sites are created by the phase_feature program. A set of pharmacophore feature definitions is applied to each ligand conformation, to identify the positions of all pharmacophoric elements in that ligand.

12.4.1 pharm_create_sites

Performs pharmacophore site creation setup and cleanup. Requires completion of project setup.

Syntax

pharm_create_sites {-setup [-fd *fdFile*]|-cleanup}

Options

-setup	Set up phase_feature job. Any forward project data will be removed upon confirmation.

`-fd` *fdFile*	Use pharmacophore feature definitions in *fdFile*. If omitted, the default definitions from the Phase installation will be used. Valid only with `-setup`.
`-cleanup`	Clean up after `phase_feature` job has finished.

Output Files

The files generated by the `-setup` option are:

`create_sites_feature.ini`	A copy of the default feature definition file, `pharma_feature.ini`.
`create_sites_phase.inp`	Main input file for `phase_feature`.

The file generated by the `-cleanup` option is:

`CreateSitesData.tab`	Summary of the pharmacophore feature counts for each ligand.

12.4.2 phase_feature

Generates pharmacophore sites for one or more ligands from a set of defined features.

Syntax

`$SCHRODINGER/phase_feature` *jobname* `[`*job-options*`]`

If you used `pharm_create_sites` to set up the job, *jobname* is `create_sites`, and the relevant input files are set up automatically.

Input Files

jobname`_phase.inp`	Phase main input file, which contains options that govern Phase behavior. See Section B.2 on page 172 for details of this file. The list of ligands should be restricted to the ligands to be used in the model development.
jobname`_feature.ini`	Feature definitions file. This file can be created from the template feature definitions file (`pharma_feature.ini`) by incorporating any changes to the standard features. The template file is located in `$SCHRODINGER/phase-v`*version*`/data`. It is strongly recommeded to edit this file using Maestro.
`mmphob.ini`	File that contains definitions for hydrophobic groups. This file is optional. Unless a local copy is supplied this file will be read from the default location in the mmshare installation.

ligand-name`.mae`.	Files containing ligand structures. These files should be stored in the ligands subdirectory, as specified in the Phase main input file. The default directory name is `ligands`. Each ligand file is a multi-conformer Maestro file. Ligand names should be listed in the Phase main input file as `LIGAND_NAME =` *ligand-name*. You should only list the active ligands to be used in the model.

Output Files

The following output files are created upon successful job completion:

jobname`_phase.log`	Log information, including the lists of mapped features for each ligand.
ligand-name`_sites.phs`	Pharmacophore site coordinates for the ligand specified by *ligand-name*. These files are created in the ligands directory.
ligand-name`_xyz.phc`	Atom coordinates of each conformer for the ligand specified by *ligand-name*. These files are created in the ligands directory, and are needed for running Phase scoring jobs. These files have a stripped-down format that allows rapid access to conformer structural data in subsequent steps of the workflow.

12.5 Finding Common Pharmacophores

Common pharmacophores are identified by the `phase_partition` program. All *n*-point pharmacophores from the active `PHARM_SET` ligands are enumerated and filtered into a set of high-dimensional boxes. Pharmacophores that fall into the same box are similar enough to be considered equivalent. Boxes that receive at least one pharmacophore from a sufficient number of actives are said to “survive” the partitioning process. Each surviving box may be processed in the subsequent Score Actives step to identify a pharmacophore hypothesis from that box. See Chapter 6 for details on the process.

12.5.1 pharm_find_common

Performs setup and cleanup for common pharmacophore perception. Requires completion of the Create Sites step.

Syntax

```
pharm_find_common -setup -sites numSites [-match minMatch] [-freq]
pharm_find_common -cleanup
```

Options

`-setup`	Set up `phase_partition` job. Any forward project data is removed upon confirmation.
`-cleanup`	Clean up after `phase_partition` job has finished. Also writes a summary of the common pharmacophore results to the file `FindCommonPharmData.tab`.
Setup Options:	
`-sites` *numSites*	The number of sites in each common pharmacophore. Must lie between 3 and 7. This option is required.
`-match` *minMatch*	The minimum number of active `PHARM_SET` ligands that must match a pharmacophore before it can be considered common. Must lie between 2 and the number of ligands in `MasterData.tab` for which `PHARM_SET = active`. If omitted, all active `PHARM_SET` ligands are required to match.
`-freq`	Use limits in `FeatureFreq.tab` to control the pool of variants considered. Individual variants can be removed from the input file `find_common_phase.inp` once the setup is complete. If this option is omitted, the minimum and maximum frequencies are 0 and 4, respectively, for all types of pharmacophore features.

Output Files

The file generated by the `-setup` option is:

`find_common_phase.inp` Phase main input file for `phase_partition`.

The file generated by the `-cleanup` option is:

`FindCommonPharmData.tab` Summary of the number of boxes for each variant.

12.5.2 phase_partition

The `phase_partition` program is used to find common pharmacophores for a given set of ligands. This step is also known as a partitioning job (from the name of underlying algorithm). This program can be run on multiple processors, which are specified with the `-HOST` option.

Syntax

`$SCHRODINGER/phase_partition` *jobname* `[`*job-options*`]`

Input Files

*jobname*_phase.inp	Phase main input file with options for this type of Phase job.
*ligand-name*_sites.phs	Site files for each of the ligands in the set, created by a phase_feature run. These files are located in the ligands directory, which is specified in the Phase main input file.
FeatureFreq.tab	Feature frequency file. Used to set mimimum and maximum allowed feature frequencies for common pharmacophore perception. See Section B.8 on page 187 for an example.

Output Files

The following intermediate and output files are created by the job in the working directory:

*jobname*_partition.inp	Partitioning input file, which is generated automatically from the Phase main input file. Used by the computational program, and is useful mainly for troubleshooting purposes.
*jobname*_partition.out	Output file. Contains some information about the job, but is mainly useful for debugging.
*jobname*_partition.log	Log file. Contains information on job progress, including boxes generated for each variant and eliminated variants.
*jobname*_partition_ variants.tab	File containing list of variants and number of boxes for each variant.
*jobname*_boxes.tar	Archive of box file archives generated by the partitioning code. Box files are archived for each variant. This archive is used by the subsequent scoring job.

12.6 Scoring Hypotheses

The Score Hypotheses stage of the workflow involves calculating scores for each possible hypothesis based on ligand alignment, volume overlap, and various properties. Only the highest-scoring hypotheses are kept. For a detailed description of how scoring is done, see Chapter 7. Scoring does not eliminate redundant hypotheses that arise from site permutations, which are treated as distinct by the partitioning algorithm. Redundancies can be identified by applying a clustering technique based on geometric similarity.

Hypotheses are scored with respect to the active PHARM_SET ligands by the program phase_scoring. This process assigns numerical rankings to the pharmacophores within each surviving box from the Find Common Pharmacophores step. The highest scoring pharmacophore in a given box is designated as a hypothesis, and the ligand giving rise to that pharmacophore is known as its reference ligand. The scoring function considers the quality of the alignments afforded by each pharmacophore, along with a number of other user-configurable factors. See Section 7.1 on page 66 for more information on the scoring process.

12.6.1 pharm_score_actives

Performs setup and cleanup for scoring of actives. Requires the completion of the Find Common Pharmacophores step.

If you modify the scoring function with any of the options listed below, you should examine the range of values of the property to choose an appropriate weight. In general, it is advisable to ensure that contributions to the scoring function are in the range 0.0 to 1.0 in magnitude, to prevent the contribution from completely dominating the scoring function.

Syntax

```
pharm_score_actives -setup [-tol] [-act weight | -prop weight]
    [-conf weight]
pharm_score_actives -cleanup
```

Options

`-setup`	Set up `phase_scoring` job. Any forward project data is removed upon confirmation.
`-cleanup`	Clean up after `phase_scoring` job has finished. Hypothesis files *hypoID*`.def`, *hypoID*`.mae`, *hypoID*`.tab`, and *hypoID*`.xyz` are created in the directory `hypotheses`, and a summary of the active scoring results is written to the files `ScoreActivesData.tab` and `ScoreActivesData.csv`.
Setup Options:	
`-tol`	Use feature matching tolerances in `FeatureTol.tab` when performing alignments. The default is to apply a single threshold to the overall RMSD.
`-act` *weight*	Incorporate reference ligand `ACTIVITY` into the scoring function, multiplied by the supplied weight. This affects the overall score, and it biases the selection of reference ligands to favor those with higher `ACTIVITY`.
`-prop` *weight*	Incorporate reference ligand `1D_VALUE` into the scoring function, multiplied by the supplied weight. This affects the overall score, and it biases the selection of reference ligands to favor those with higher `1D_VALUE`.

`-conf` *weight*	Incoporate reference ligand relative conformational energy into the scoring function, multiplied by the supplied weight (positive weights are negated automatically). This affects only the overall score, not the selection of reference ligands. Valid only if `-conf` *confProp* was used when `pharm_project` was run.

Output Files

The following files are created with the `-setup` option:

`score_actives_phase.inp`	Phase main input file.
`score_actives_feature.ini`	Feature definitions file, as provided to the prior `phase_feature` job.
`score_actives_boxes.tar`	Copy of the archive of box files generated by the `phase_partition` job.

The following files are generated with the `-cleanup` option.

`ScoreActivesData.tab`	Plain text summary of results in tabular form.
`ScoreActivesData.csv`	Summary of results in comma-separated value form.
hypoID`.def`	Feature definitions for the given hypothesis. Stored in the `hypotheses` subdirectory.
hypoID`.mae`	Maestro format file containing aligned actives for the given hypothesis. Stored in the `hypotheses` subdirectory.
hypoID`.tab`	Primary hypothesis data for the given hypothesis. Stored in the `hypotheses` subdirectory.
hypoID`.xyz`	Site coordinates for the given hypothesis. Stored in the `hypotheses` subdirectory.

12.6.2 phase_scoring

Scores and ranks pharmacophore hypotheses for actives. This program can be run on multiple processors, which are specified with the `-HOST` option.

Syntax

`$SCHRODINGER/phase_scoring` *jobname* `[`*job-options*`]`

Input Files

*jobname*_phase.inp	Phase main input file.
*jobname*_feature.ini	Feature definitions file, as provided to the phase_feature job.
mmphob.ini	Hydrophobic groups definitions file, as provided to the phase_feature job.
*jobname*_boxes.tar	Archive of box files generated by the phase_partition job. Box files are archived for each variant. This archive is expanded internally during execution.
ligand-name.mae	Files containing ligand structures, in the ligands directory, as provided to the phase_feature job.
*ligand-name*_sites.phs	Pharmacophore site coordinate files for each ligand, in the ligands directory, as generated by phase_feature.
*ligand-name*_xyz.phc	Ligand conformation files, in the ligands directory, as generated by phase_feature.
FeatureTol.tab	Feature-matching tolerances file. Optional.

Output Files

The following output files are generated by this job:

*jobname*_scoring.log	Log file. Contains information on job progress. Stored in the current directory.
*jobname*_scoring.tar	Archive file that contains all the results of the scoring job. Stored in the current directory.
*jobname_variant*_hypothesis.tab	File containing a list of hypotheses for the given variant, ordered according to hypothesis rank. Stored in the archive file.
*jobname_variant*_scores.out	File containing information about hypotheses for each variant. *variant-name* is encoded from the variant name as a string of integers; for example AADDH is encoded as 00112. Stored in the archive file.
*variant*_str_*N*.mae	Maestro format file containing ligand structures aligned onto the reference ligand for a given hypothesis. *N* is the unique identifier of the box from which this hypothesis came. Stored in the archive file.
*variant*_hyp_*N*.xyz	Site coordinate information file for the given hypothesis. Stored in the archive file.

12.6.3 pharm_score_inactives

Performs setup and cleanup for scoring of inactives. Requires the completion of the `pharm_score_actives` step.

Hypotheses are scored with respect to inactive `PHARM_SET` molecules by the program `phase_inactive`. Survival scores are adjusted to penalize hypotheses that consistently match inactives. The premise behind this technique is that the inactives fail to bind because they do not contain the true pharmacophore. While this condition is rarely satisfied by every inactive in a given dataset, it is important that at least some significant fraction of the inactives lack the pharmacophore. See Section 7.3 on page 71 for more information.

Syntax

`pharm_score_inactives {-setup -w` *weight* | `-cleanup}`

Options

`-setup`	Set up `phase_inactive` job. Any forward project data is removed upon confirmation.
`-w` *weight*	Survival scores from the Score Actives step are adjusted by subtracting this weight multiplied by the average fitness obtained for the molecules in the inactive `PHARM_SET`.
`-cleanup`	Cleanup after `phase_inactive` job has finished. Writes a summary of the inactive scoring results to the files `ScoreInactivesData.tab` and `ScoreInactivesData.csv`.

Output Files

The following files are created with the `-setup` option:

`score_inactives_phase.inp`	Phase main input file.
`score_inactives_inactive.inp`	Input file for `phase_inactive` (see Section B.4 on page 179).
`score_inactives_feature.ini`	Feature definitions file, as provided to the prior `phase_feature` job.
`score_inactives_hypoFiles.tar`	Archive of hypothesis files.
`score_inactives_ligandFiles.tar`	Archive of inactive ligand structure files.

12.6.4 phase_inactive

Scores pharmacophore hypotheses for inactives. This program can be run on multiple processors, which are specified with the `-HOST` option (see Section 12.1 on page 109).

Syntax

`$SCHRODINGER/phase_inactive` *jobname* `[`*job-options*`]`

Input Files

Uses the same input files as `phase_scoring`, and the following file in addition. The main input file must list only the inactives in the `LIGAND_NAME` records. This is done automatically by `pharm_score_inactives`.

jobname`_inactive.inp` Phase inactives input file.

Output Files

The following output files are generated by this job:

jobname`_inactive.log`	Log file. Contains information on job progress. Stored in the current directory.
`ScoreInactivesData.tab`	Plain text summary of results in tabular form.
`ScoreInactivesData.csv`	Summary of results in comma-separated value form.

12.6.5 pharm_cluster_hypotheses

Performs setup and cleanup for the clustering of hypotheses. Clustering is performed by the program `phase_hypoCluster`. Requires completion of the `pharm_score_actives` step.

Syntax

```
pharm_cluster_hypotheses {-setup [-link method] | -cleanup }
    -report level
```

Options

`-setup`	Set up `phase_hypoCluster` job. Any forward project data is removed upon confirmation.
`-link` *method*	Cluster linkage method. Allowed values are as follows: `single` Use the highest similarity between any two objects from the two clusters. Produces diffuse, elongated clusters. `average` Use the average similarity between all pairs of objects from the two clusters. `complete` Use the lowest similarity between any two objects from the two clusters. Produces compact, spherical clusters. The default is `complete`.
`-cleanup`	Cleanup after `phase_hypoCluster` job has finished.
`-report` *level*	Report results for a particular level of clustering. If *level* lies between 0 and 1, it is treated as a merge similarity, and the clusters reported will correspond to the point at which the merge similarity becomes less than or equal to *level*. If *level* is 2 or greater, it is treated as a cluster count, and results are reported for the formation of *level* clusters. Different values of *level* can be tried without rerunning `phase_hypoCluster`. To determine an appropriate value for *level*, examine the file `cluster_hypothesis_hypoCluster.log`.

Output Files

The following files are created with the `-setup` option:

`cluster_hypotheses_phase.inp`	Phase main input file.
`cluster_hypotheses_hypoCluster.inp`	Input file for `phase_hypoCluster` (see Section B.5 on page 181).
`cluster_hypotheses_feature.ini`	Feature definitions file used to create hypotheses.
`cluster_hypotheses_hypoFiles.tar`	Archive of hypothesis files.

The following files are generated with the `-cleanup` option.

`ClusterHypothesesData.tab`	Plain text summary of results in tabular form.
`ClusterHypothesesData.csv`	Summary of results in comma-separated value form.

12.6.6 phase_hypoCluster

Hierarchical agglomerative clustering of hypotheses is performed by `phase_hypoCluster`, using a geometric similarity computed from the least-squares alignment of each pair of hypotheses *i*, *j*:

$$\mathrm{Sim}(i, j) = \frac{\langle i \mid j \rangle}{\sqrt{\langle i \mid i \rangle \langle j \mid j \rangle}}$$

where

$$\langle i \mid j \rangle = S_{\mathrm{site}}(i, j)\, W_{\mathrm{align}} + S_{\mathrm{vec}}(i, j)\, W_{\mathrm{vec}}$$

The site and vector scores are computed just as in the Score Actives step (see Section 7.1 on page 66). When more than one mapping is possible, the alignment yielding the highest similarity is used. Hypotheses that do not contain the same pharmacophore features (i.e., different variants) are assigned a similarity of zero.

Syntax

`$SCHRODINGER/phase_hypoCluster` *jobname* `[`*job-options*`]`

Input Files

jobname`_hypoCluster.inp` Hypothesis clustering input file.

The remainder of the input files are specified in the hypothesis clustering input file—see Section B.5 on page 181.

Output Files

jobname`_hypoCluster.log` Log file. Contains information on job progress and results in readable form. Stored in the current directory.

The name of the output file containing the results of the cluster analysis is specified in the hypothesis clustering input file—see Section B.5 on page 181.

12.7 Building QSAR Models

The Build QSAR Model step develops a QSAR model based on partial least-squares (PLS) analysis for one or more hypotheses. The ligands are aligned to each hypothesis as part of the process. For more information on the QSAR model, see Section 8.1 on page 77.

12.7.1 pharm_build_qsar

Performs setup and cleanup for building a 3D QSAR model. Requires the completion of the `pharm_score_actives` step.

A QSAR model is created for each hypothesis by the program `phase_multiQsar`.

Syntax:

`pharm_build_qsar {-setup [`*options*`] | -cleanup}`

Options:

`-setup`	Set up `phase_multiQsar` job. Any forward project data is removed upon confirmation by the user.
`-cleanup`	Clean up after `phase_multiQsar` job has finished. Writes a summary of the QSAR statistics to the files `BuildQsarData.tab` and `BuildQsarData.csv`. Updated hypotheses and input files for `phase_qsar` jobs are written to the directory `BuildQsarResults`.
Setup Options:	
`-model` *type*	Model type. Allowed values are `atom` and `pharm`. Atom-based models consider the space occupied by all atoms in each molecule, whereas pharmacophore-based models consider only the space occupied by pharmacophore sites that match the hypothesis. Default: `atom`..
`-grid` *spacing*	Grid spacing in angstroms. Must lie between 0.5 and 4.0. The default and recommended value is 1.0.
`-factors` *n*	The maximum number of PLS factors in each model. The default and largest legal value is 1/5 the number of training set molecules.
`-atomVol`	Consider only atoms of the same MacroModel type when computing volume score overlaps during alignment. This favors alignments that superimpose chemically similar atoms. The default is to ignore atom types when computing volume scores.

Output Files

The following files are created with the `-setup` option:

`build_qsar_multiQsar.inp`	Input file for `phase_multiQsar` (see Section B.6 on page 182).
`build_qsar_hypoFiles.tar`	Archive of hypothesis files.
`build_qsar_ligandFiles.tar`	Archive of inactive ligand structure files.

The `-cleanup` option extracts the results archive from the `phase_multiQsar` job into the directory `BuildQsarResults`. These files are listed in the next section.

12.7.2 phase_multiQSAR

This is a driver program that builds QSAR models for multiple hypothesis, by running `phase_qsar` on individual hypotheses and collecting the results.

Syntax

`phase_multiQsar [`*job-options*`]` *jobname*

Input Files

jobname`_multiQsar.inp`	Multiple QSAR model input file.

The remainder of the input files are specified in the multiple QSAR model input file—see Section B.7 on page 185.

Output Files

jobname`_multiQsar.log`	Log file. Contains information on job progress.
jobname`_multiQsar.tar`	Archive file (`.tar`). Contains files relating to QSAR models.
`BuildQsarData.tab`	Plain text summary of results in tabular form.
`BuildQsarData.csv`	Summary of results in comma-separated value form.

Files for each hypothesis are stored in the subdirectory specified by the `resultDir` keyword in the input file, which is set to `BuildQsarResults` by `pharm_build_qsar`, inside the archive file. These files are listed below.

*hypoID*_align.mae	Aligned structures for training and test set molecules that matched at least 3 sites in the hypothesis. Training set molecules appear first. Stored in archive file.
hypoID.def	Feature definitions (copied from input). Stored in archive file.
hypoID.mae	Reference ligand structure (copied from input). Stored in archive file.
*hypoID*_order.dat	File that defines the overall order of molecules in *hypoID*_align.mae.
*hypoID*_pharm.dat	The pharmFile required by phase_qsar, if a pharmacophore-based model was chosen (see Section B.7 on page 185).
hypoID.qsar	QSAR model file.
*hypoID*_qsar.inp	Main input file for phase_qsar job.
hypoID.rad	Copy of the feature radius file, if specified in the input file.
hypoID.tab	Primary hypothesis data, with QSAR model flag activated.
hypoID.tol	Copy of the feature cutoff file, if specified in the input file.
hypoID.xyz	Hypothesis site coordinates (copied from input).

12.7.3 phase_qsar

The phase_qsar program creates and applies grid-based 3D QSAR models. It makes activity predictions and generates detailed output for individual QSAR models for a single hypothesis. If pharmFile is specified in the input file, a feature-based QSAR model is developed or tested rather than an atom-based model. You can generate a pharmFile by running phase_fileSearch on the Maestro file that contains the molecules of interest, with pharmFile specified in the input file to phase_fileSearch.

Syntax

```
$SCHRODINGER/phase_qsar jobname
```

Input Files

*jobname*_qsar.inp	QSAR model input file.

The remainder of the input files are specified in the QSAR model input file—see Section B.7 on page 185.

Output Files

*jobname*_qsar.log	Log file. Contains information on job progress.
*jobname*_qsar.out	Output file. Contains complete model statistics, Cartesian coordinates and regression coefficient for each bit in the model, training and test set predictions, and actual bit values for each molecule.

The remainder of the output files are specified in the QSAR model input file—see Section B.7 on page 185.

12.7.4 phase_qsar_stats

Extract statistics from Phase QSAR models and from hit files that contain QSAR predictions.

Syntax

phase_qsar_stats -hypo *hypoID* [-hits *hitFile* [-act *actProp*] [-plot *csvFile*]]
    [-out *outFile*]

Options

-hypo *hypoID*	Prefix that identifies the hypothesis from which the QSAR model was derived. The file *hypoID*.qsar must be present.
-hits *hitFile*	Maestro file containing hits that match the hypothesis from which the QSAR model was derived.
-act *actProp*	Experimental activity property name exactly as it appears in *hitFile*.
-plot *csvFile*	Comma-separated value file for output of experimental and predicted activities.
-out *outFile*	File for program output. If omitted, standard output is used.

12.7.5 qsarVis

Visualize QSAR models that you create in command line projects. This utility launches a graphical interface entitled Visualization Tookit – OpenGL, with an interactive 3D image of the ligand, hypothesis, and QSAR model. This interface displays the QSAR model in a similar way to the Maestro interface—see Section 8.5 on page 84 for more information.

You can rotate, translate, and zoom in using the mouse, but the controls are different from those in Maestro. To rotate the image, drag with the left mouse button; to translate, drag with the middle mouse button; to zoom, drag with the right mouse button.

To change the visualization settings, you must start a new instance of `qsarVis`. However, if you place each instance in the background, you can display and compare QSAR models with various settings.

Note: This utility is only available on Linux-x86 platforms.

Syntax

`qsarVis -hyp` *hypoID* `-mol` *molname* `[`*options*`]`

Options

`-hyp` *hypoID*	Hypothesis ID. Required.
`-mol` *molname*	Molecule name (for example, mol_5). Required.
`-volume_qsar`	View volume bits in QSAR model.
`-class` *name*	View effects from a single atom/feature class. Allowed values are D, H, N, P, W, X.
`-pc` *posThresh*	Threshold for display of positive values. Default: 0.02.
`-nc` *negThresh*	Threshold for display of negative values. Default: –0.02
`-trans` *value*	Transparency value. Allowed values are between 0.0 and 1.0. Default: 0.5.
`-npls` *plsFactors*	Number of PLS factors. Default: 1.

12.8 Adding Excluded Volumes to a Hypothesis

12.8.1 create_xvolShell

Creates a shell of excluded volume spheres to surround the reference ligand of a pharmacophore hypothesis. This shell defines the outer boundary of a shape-based constraint that can be applied when searching for matches to the hypothesis.

Syntax

`create_xvolShell -hypo` *hypoID* `[-ref` *maeFile*`] [-buff` *dist*`] [-grid` *spacing*`]`
`[-hydrogens] [-append] [-cut]`

Options

`-hypo` *hypoID*	File prefix for hypothesis. Required. The files *hypoID*`.mae` and *hypoID*`.tab` must be present (unless `-ref` *maeFile* is used). The file *hypoID*`.xvol` is created with the excluded volume data in a format that is recognized by both the Phase GUI and Phase programs.
`-ref` *maeFile*	Build the shell around the structures in *maeFile*. By default, the reference conformer in *hypoID*`.mae` is used.
`-buff` *dist*	Buffer distance in angstroms between the excluded volume surface and the van der Waals surface of the reference ligand. Default: 1.0.
`-grid` *spacing*	The size in angstroms of the grid used to assign excluded volume sphere positions. Also determines the sphere radii. Default: 1.0.
`-hydrogens`	Consider hydrogens when creating the shell and when checking for excluded volume violations. If `-hydrogens` is used, then the file *hypoID*`.ev` is created with excluded volume data in a format that is recognized only by Phase computational programs. This file contains a special flag that signals the programs to consider hydrogens when checking for excluded volume violations. The Phase GUI does not support the use of this option, so the search must be set up and run through the command line using *hypoID*`.ev` as the source of excluded volumes. Default: consider only non-hydrogen atoms.
`-append`	Append to existing excluded volumes. If this option is used, excluded volumes are added to existing volumes stored in either *hypoID*`.ev`, or *hypoID*`.xvol`. If both files exist, *hypoID*`.ev` will be used.
`-cut`	Create excluded volumes with a cutaway view. Since the shell of spheres typically obscures the view of the reference ligand, this option is provided to allow creation of only half the shell. The hypothesis and cutaway excluded volumes may then be imported into the Phase GUI to confirm that the shell is surrounding the ligand as expected. If everything is satisfactory, this program should be rerun without the `-cut` option.

12.8.2 create_xvolClash

Creates excluded volumes using actives and inactives that have been prealigned to a pharmacophore hypothesis. Excluded volumes are placed in locations that would cause steric clashes only for the inactives.

Syntax

```
create_xvolClash -hypo hypoID -pos maeFilePos -neg maeFileNeg
    [-freq minClash] [-buff dist] [-grid spacing] [-hydrogens] [-append]
```

Options

`-hypo` *hypoID*	File prefix for hypothesis. Required. The file *hypoID*`.xvol` is created with the excluded volume data in a format that is recognized by both the Phase GUI and Phase programs.
`-pos` *maeFilePos*	Maestro file containing the actives, aligned to the hypothesis. Required.
`-neg` *maeFileNeg*	Maestro file containing the inactives, aligned to the hypothesis. Required.
`-freq` *minClash*	The minimum number of inactives that must experience a clash before creating an excluded volume sphere. Default: 1.
`-buff` *dist*	Buffer distance in angstroms between the excluded volume surface and the van der Waals surface of the reference ligand. Default: 1.0.
`-grid` *spacing*	The size in angstroms of the grid used to assign excluded volume sphere positions. Also determines the sphere radii. Default: 1.0.
`-hydrogens`	Consider hydrogens when creating the shell and when checking for excluded volume violations. If `-hydrogens` is used, then the file *hypoID*`.ev` is created with excluded volume data in a format that is recognized only by Phase computational programs. This file contains a special flag that signals the programs to consider hydrogens when checking for excluded volume violations. The Phase GUI does not support the use of this option, so the search must be set up and run through the command line using *hypoID*`.ev` as the source of excluded volumes. Default: consider only non-hydrogen atoms.
`-append`	Append to existing excluded volumes. If this option is used, excluded volumes are added to existing volumes stored in either *hypoID*`.ev`, or *hypoID*`.xvol`. If both files exist, *hypoID*`.ev` will be used.
`-cut`	Create excluded volumes with a cutaway view. Since the shell of spheres typically obscures the view of the reference ligand, this option is provided to allow creation of only half the shell. The hypothesis and cutaway excluded volumes may then be imported into the Phase GUI to confirm that the shell is surrounding the ligand as expected. If everything is satisfactory, this program should be rerun without the `-cut` option.

12.8.3 create_xvolReceptor

Creates excluded volumes from a receptor structure or a portion of a receptor structure. An excluded volume sphere is created for each atom in the receptor structure that that satisfies the minimum and maximum distance criteria.

Syntax

`create_xvolReceptor -hypo` *hypoID* `-receptor` *maeFile* `[-radius` *r*`]`
`[-buff` *dmin*`] [-limit` *dmax*`] [-hydrogens] [-append]`

Options

`-hypo` *hypoID*	File prefix for hypothesis. Required. The file *hypoID*`.xvol` is created with the excluded volume data in a format that is recognized by both the Phase GUI and Phase computational programs.
`-receptor` *maeFile*	Maestro file containing the receptor structure. Required.
`-radius` *r*	Radius for excluded volume spheres. Default: use the van der Waals radius of each receptor atom.
`-buff` *dmin*	Buffer distance in angstroms between the excluded volume surface and the van der Waals surface of the reference ligand. Default: 1.0.
`-limit` *dmax*	Limit the thickness of the shell created by ignoring receptor atoms that are more than a distance *dmax* from the reference ligand. By default, no limit is applied.
`-hydrogens`	Consider hydrogens when creating the shell and when checking for excluded volume violations. If `-hydrogens` is used, then the file *hypoID*`.ev` is created with excluded volume data in a format that is recognized only by Phase computational programs. This file contains a special flag that signals the programs to consider hydrogens when checking for excluded volume violations. The Phase GUI does not support the use of this option, so the search must be set up and run through the command line using *hypoID*`.ev` as the source of excluded volumes. Default: consider only non-hydrogen atoms.
`-append`	Append to existing excluded volumes. If this option is used, excluded volumes are added to existing volumes stored in either *hypoID*`.ev`, or *hypoID*`.xvol`. If both files exist, *hypoID*`.ev` will be used.

12.9 Other Utilities

12.9.1 pharm_archive

Archives forward steps in a Phase pharmacophore model project using `tar` and `gzip`, allowing data to be preserved before it is overwritten when a step is rerun.

Syntax

`pharm_archive -step` *stepName* `-tar` *tarFile* `[-gzip]`

Options

`-step` *stepName*	The step at which to begin archiving. Allowed values of *stepName* are listed below, with the step number and the steps archived:

1	`project`	Entire project
2	`create_sites`	2–7
3	`find_common`	3–7
4	`score_actives`	4–7
5	`score_inactives`	5
6	`cluster_hypotheses`	6
7	`build_qsar`	7

`-tar` *tarFile*	Name of `tar` archive to be created.
`-gzip`	After creating archive, compress using `gzip`.

12.9.2 pharm_align_mol

Performs setup and cleanup for aligning molecules to a pharmacophore hypothesis. Supports alignment of command-line project ligands and structures stored in a Maestro or SD file.

Alignments are generated by the program `phase_fileSearch`, which can operate on existing conformers, or generate them during the search. See Section 14.1 on page 153 for information on this program.

Syntax

`pharm_align_mol -setup [`*options*`] | -cleanup -job` *jobName*

Options

`-setup`	Set up `phase_fileSearch` job. Existing job files that would be overwritten are removed upon confirmation.
`-cleanup`	Clean up after `phase_fileSearch` job has finished.
`-job` *jobName*	`phase_fileSearch` job name. Required for setup and cleanup.
Setup Options:	
`-hypo` *hypoID*	Hypothesis ID. This is the prefix for the associated hypothesis files (*hypoID*`.tab`, *hypoID*`.xyz`, etc.) Include path if these files are not in the current directory.
`-lig` *recordFile*	File containing `LIGAND_NAME` records for the subset of project ligands to be aligned. Use `ProjectLigands.inp` if you wish to align all ligands. Otherwise, copy `ProjectLigands.inp` to *recordFile*, and delete or comment out `LIGAND_NAME` records you wish to skip. You must run the job from the project directory when using this option.
`-mol` *structFile*	Maestro or SD file containing cleaned 3D structures to be aligned. Successive structures with the same title and connectivity are treated as conformers of a single molecule unless `-flex` is used.
`-flex` *maxConfs*	Generate up to `maxConfs` conformers during the search, for each structure in *structFile*. Do not use this option if *structFile* already contains multiple conformers per molecule.
`-match` *minSites*	The minimum number of hypothesis sites that must be matched. Must lie between 3 and the number of sites in the hypothesis. The default is to match all sites.
`-phase` *inpFile*	Phase-style input file from a completed project step involving ligand alignments. The files for each step are listed below:

Score Actives	`score_actives_phase.inp`
Score Inactives	`score_inactives_phase.inp`
Cluster Hypotheses	`cluster_hypotheses_phase.inp`
Build QSAR	`build_qsar_phase.inp`

If this option is used, the `phase_fileSearch` alignment options will be consistent with those used in the applicable project step. If this option is omitted, default alignment options are written to the `phase_fileSearch` input file.

12.9.3 align_hypoPair

Aligns one hypothesis onto another. Alignment is done using least-squares fitting of the matching site points in the two hypotheses, considering all possible mappings. Alignments are summarized to standard output in order of increasing RMSD. By default, only the best alignment is saved as a new hypothesis, but this can be overridden.

Syntax

`align_hypoPair -fixed` *fixedHypoID* `-free` *freeHypoID* `-new` *newHypoID*
`[-dtol` *deltaDist*`] [-match` *minSites*`] [-mix] [-equiv` *equivFile*`]`
`[-merge` *method*`] [-keep` *maxAlign*`] [-rmsd` *rmsdFile*`]`

Options

`-fixed` *fixedHypoID*	File prefix for the hypothesis that remains fixed when the alignment is performed. At a minimum, the files *fixedHypoID*`.xyz` and *fixedHypoID*`.def` must be present. Matching filters are applied if the associated files are present: *fixedHypoID*`.tol` for feature-matching tolerances, *fixedHypoID*`.dxyz` for hypothesis-specific tolerances, *fixedHypoID*`.mask` for site masks.
`-free` *freeHypoID*	File prefix for the hypothesis that will be aligned to the fixed hypothesis (the “free” hypothesis). At a minimum, the files *freeHypoID*`.xyz` and *freeHypoID*`.def` must be present. The feature definitions in *freeHypoID*`.def` must be identical to those in *fixedHypoID*`.def`.
`-new` *newHypoID*	File prefix for the aligned version of the free hypothesis. The files *newHypoID*`.def` and *newHypoID*`.xyz` are created automatically. The files *newHypoID*`.mae` and *newHypoID*`.tab` are created if *freeHypoID*`.mae` and *freeHypoID*`.tab` are present, except when `-merge` is used.
`-dtol` *deltaDist*	Intersite distance matching tolerance. Default: 2.0 Å.
`-match` *minSites*	Minimum number of sites that must match. Default: 3.
`-mix`	Consider alignments involving different numbers of matching sites. By default, only *n*-point matches are retained and ranked by RMSD, where *n* is the greatest number of sites that could be matched. If `-mix` is specified, matches involving fewer sites will be mixed with the *n*-point matches. RMSD values are generally smaller for matches with fewer sites, so using the `-mix` option favors those alignments.

`-equiv` *equivFile*
: File that defines the allowed mappings between the sites in the fixed and free hypotheses. This file consists of two lines: the first for the fixed hypothesis and the second for the free hypothesis. Each line contains a string of arbitrary non-blank characters; each string contains one character for each site. The mapping is defined by choosing the same character in each string for sites that match.

For example, if the fixed hypothesis contains 4 sites and the free hypothesis contains 5 sites, the following strings could be used to define the allowed mappings:

```
abbc
ababc
```

In this mapping, the first site in the fixed hypothesis, denoted as `a`, can be matched to either the first or third site of the free hypothesis, because these are also denoted as `a`. The second and third sites of the fixed hypothesis, denoted as `b`, can be matched to the second or fourth sites of the free hypothesis. The fourth site of the fixed hypothesis, denoted as `c`, can be matched to only the fifth site of the free hypothesis.

If this option is omitted, the variants for the two hypotheses are used to define the allowed mappings, hence only sites of the same type can be matched to each other.

`-merge` *method*
: Merging method. If this option is used, the new hypothesis is a union of the sites in the fixed and aligned free hypotheses, with matching sites merged or replaced as follows:

method	Merged Site
1	Fixed site type; fixed site coordinates
2	Free site type; free site coordinates
3	Fixed site type; average coordinates
4	Free site type; average coordinates

When merging is done, the new hypothesis has no real reference ligand, so the files *newHypoID*`.mae` and *newHypoID*`.tab` are not created. The RMSD value is computed based on the positions of the aligned free sites, not the positions of the merged sites.

`-keep` *maxAlign*	Maximum number of aligned hypotheses to keep. By default, only a single set of hypothesis files is created with the prefix *newHypoID*, corresponding to the smallest RMSD. However, if *maxAlign* is greater than 1 and multiple alignments are possible, a series of hypotheses *newHypoID*`_1`, *newHypoID*`_2`, ... is created, with progressively larger RMSD values.
`-rmsd` *rmsdFile*	File to which RMSD values should be written, one per line. The file contains one value for each hypothesis created, up to *maxAlign*. By default, no RMSD file is written.

Managing and Searching 3D Databases from the Command Line

Phase provides two ways of searching a set of structures for matches to a hypothesis: searching a plain Maestro file, or creating a database for searching. In both cases, the process involves generating conformers of each structure and creating the site points. For a search on a Maestro file, the conformers and sites are generated during the search. When a database is created, the conformers and the sites can be stored in the database. In this case, the search step only involves matching the sites to the hypothesis, which is much quicker than the conformational search. So if you intend to search a set of structures more than once with the same set of features, you should consider creating a database and storing conformers in the database. Another advantage of creating a database is that you can define database subsets, which can be searched.

The structures that are used for searching must be all-atom, 3D structures in the correct ionization state. If the structures are not in this form—for example, if they are 2D structures—they must be prepared in the correct form first, which you can do with LigPrep. See the *LigPrep User Manual* for details. If you already have a database from some other source that matches these requirements, you can use it for Phase searches by exporting it to an SD file.

The structures in the database can be stored in Maestro format or in SD format, but only one format is used for a given database, and is chosen when the database is created. The input files can be in either format. You can convert between SD files and Maestro files with the utility `sdconvert`:

`$SCHRODINGER/utilities/sdconvert -isd` *SD-file* `-omae` *Maestro-file*

The database must be stored on a file system that is accessible to all hosts that will be used to create and to search the database.

Phase provides a set of tools for managing and searching 3D databases from the command line. These tools are stored in `$SCHRODINGER/utilities`, and are prefixed with `phasedb_`. Each section in this chapter describes the use of one of these tools. Some of the tools perform setup and cleanup for Phase computational programs, which are stored in `$SCHRODINGER`. Information on searching a plain Maestro file from the command line is given in Chapter 14.

All of the tools can be run as a job under Job Control, and each tool accepts the job options listed in Table 13.1.

Table 13.1. Job options for 3D database jobs

Option	Description
`-JOB` *jobname*	Job name. If omitted, no other job control options are permitted.
`-HOST` *host*	Run job on the specified host. To use multiple CPUs on a given host, append a colon and the number of processors after the host name: for example `-HOST myhost:4`. To use multiple hosts, enclose the blank-separated list of hosts in quotes: for example, `-HOST "cluster:8 myhost yourhost"`. Not all 3D database jobs accept multiple hosts.
`-LOCAL`	Store temporary job files in current directory.
`-TMPDIR` *dir*	Store temporary job files in *dir*.
`-WAIT`	Do not return control to the shell until job finishes.
`-INTERVAL` *n*	Update log file every *n* seconds.
`-NICE`	Run job at reduced priority.

13.1 Managing a 3D Database: phasedb_manage

Once you have a set of structures that satisfy the requirements listed above, you can set up the Phase database, add ligands to the database, and remove ligands from the database. If your set of structures exists as several files, you can create the Phase database with the first file, then add the others.

In addition to adding molecules to the database, the feature definition file used to create sites for the molecules is copied into the database. This is normally done when the database is created, but you can subsequently copy a new feature definition file into the database with `phasedb_manage`. However, if you do so after creating sites with `phasedb_confsites`, you should ensure that you create sites again for all molecules in the database to ensure that the site definitions are consistent.

The feature definition file does not have to correspond to any particular hypothesis. In general, you should create the database using the most universal set of feature definitions that you have, so that you do not have to re-create the sites in the database for different hypotheses. However, if you want to search the database with a hypothesis that was created using feature definitions that differ from those used to create the database, the sites can be created as needed when the database is searched.

The `phasedb_manage` utility is the primary tool for creating and modifying Phase 3D databases. It may be run as a regular foreground process, or as a single-CPU job on any host that has access to the database directory. The general syntax of the command is as follows:

`$SCHRODINGER/utilities/phasedb_manage [`*job-options*`] [`*database-options*`]`

The job options are described in Table 13.1, and the database options are described in Table 13.2. The detailed syntax is best described via usage cases, since only certain combinations of options are permitted.

To create a new database:

`phasedb_manage -new` *dbFileType* `[-fd` *fdFile*`] -db` *dbname* `{-mae` *maeFile*`|`
`-sd` *sdFile*`} -confs` *multiConfs* `[-ignoreTitles]`

If you want to use nonstandard or custom feature definitions, specify the feature definition file with the `-fd` option.

To add molecules to an existing database:

`phasedb_manage -add -db` *dbname* `{-mae` *maeFile*`|-sd` *sdFile*`} -confs` *multiConfs*
`[-ignoreTitles]`

To delete molecules from a database:

`phasedb_manage -delete -db` *dbname* `-records` *recordFile*

To competely remove a database:

`phasedb_manage -delete -db` *dbname* `-all`

Table 13.2. Database options for phasedb_manage

Option	Description
`-new` *dbFileType*	Create a new database using the specified storage format. *dbFileType* must be `mae` or `sd`. The storage format applies to all subsequent additions.
`-fd` *fdFile*	Use pharmacophore feature definitions in *fdFile*. If this option is omitted, the default definitions in the Phase installation are used.
`-db` *dbname*	Database name, including absolute path.
`-mae` *maeFile*	Maestro file containing cleaned structures to store in the database. If running as a job, use the absolute path to avoid file copy.
`-sd` *sdFile*	SD file containing cleaned structures to store in database. If running as a job, use the absolute path to avoid file copy.
`-confs` *multiConfs*	Option indicating whether or not the supplied structure file contains multiple conformations per molecule. *multiConfs* must be `true` or `false`. If `true`, consecutive structures with identical titles and connectivities are treated as conformations of a single molecule. If `false`, each structure is treated as a separate molecule.

Table 13.2. Database options for phasedb_manage

Option	Description
`-ignoreTitles`	Ignore titles when perceiving conformations. If this option is used, consecutive structures with identical connectivities are treated as conformations of a single molecule, even if their titles differ. Valid only when *multiConfs* is `true`.
`-records` *recordFile*	Text file containing list of records to delete. Each line in the file should have the form: `LIGAND_NAME =` *ligandName* `#` *optional comments* where *ligandName*`.mae.gz` or *ligandName*`.sdf.gz` is one of the compressed structure files in the database subdirectory *dbname*`_ligands`. If present, the pound sign `#` and any text following it is ignored.

13.2 Generating Conformers and Sites: phasedb_confsites

The `phasedb_confsites` utility creates conformations and pharmacophore sites in a Phase 3D database. Sites are always created, but conformation creation is optional. The feature definitions used for the sites are taken from the definition file copied into the database at the point it was created. If you want to use custom feature definitions, you can copy the feature definition file into the database by running `phasedb_manage` with the `-fd` option. You should do this when you create the database or before you run `phasedb_confsites` for the first time. If you change the feature definitions after creating sites, you risk embedding inconsistencies in the database, and you should run `phasedb_confsites` on all molecules in the database to ensure that there are no inconsistencies.

By default, sites are created for all molecules. If you have already created sites for some of the molecules in the database and want to create sites only for those molecules that do not have them, you should create a subset by running `phasedb_subset` with the `-sites false` option (see Section 13.4 on page 150), and using this subset to restrict the range of molecules for site creation.

The command syntax is as follows:

`$SCHRODINGER/utilities/phasedb_confsites -db` *dbname* `-JOB` *jobname*
`[`*job-options*`] [`*database-options*`]`

The job options are described in Table 13.1, and the database options are described in Table 13.3. The `-JOB` *jobname* job option is required. You can run `phasedb_confsites` on multiple CPUs.

Table 13.3. Database options for phasedb_confsites

Option	Description
`-db` *dbName*	Database name, including absolute path. Required.
`-confs` *mode*	Generate conformations using torsional sampling. Allowed values of mode are: `all` Generate conformations for all molecules. `auto` Generate conformations only for molecules that do not already have them. If this option is omitted, the database structures will not be modified. This option must be used if any other conformational options (`-sample`, `-max`, `-ewin`) are used.
`-sample` *method*	Conformational sampling method. Requires `-confs` option. Allowed values of method are `rapid` and `thorough`. See Section 4.3.2 on page 44 for a definition of these methods. Default: `rapid`.
`-max` *maxConfs*	The maximum number of conformations to generate for each molecule. Requires `-confs` option. Default: 100.
`-ewin` *deltaE*	Conformational energy window in kcal/mol. Conformations that are higher in energy than the lowest-energy conformation by this amount are discarded. Requires `-confs` option. Default: 10.
`-sub` *dbSubset*	Operate on only a subset of the database. The file *dbSubset*`_phase.inp` must contain the applicable `LIGAND_NAME` records with the following format: `LIGAND_NAME =` *ligandName* `# optional comments` where *ligandName*`.mae.gz` or *ligandName*`.sdf.gz` is one of the compressed structure files in the database subdirectory *dbName*`_ligands`. If present, the pound sign `#` and any text following it is ignored. You can create subsets with `phasedb_subset`—see Section 13.4 on page 150.

The pharmacophore sites files (*ligandName*`.phs`) are stored in the subdirectory *dbName*`_ligands`. These files hold the coordinates of all pharmacophore features for all conformations of each molecule.

The conformers are stored in *ligandName*`.mae.gz` or *ligandName*`.sdf.gz` in the database subdirectory *dbName*`_ligands`. Each file contains all the conformers for the molecule.

13.3 Searching for Matches in a Database: phasedb_findmatches and phase_dbsearch

Database searching takes place in two steps, finding matches and fetching hits. In the find step, the database is searched for geometric arrangements of pharmacophore sites that match the feature types and intersite distances of the hypothesis, and a file containing all the matches is written out. The find step is the most expensive, and need only be run once. In the fetch step, the match file is used as a lookup table to rapidly retrieve conformations from the database and align them to the hypothesis. The fetch step can be used to filter these hits using excluded volumes, a scoring function, and numerical limits. For more information on searching the database, see Chapter 11.

The utility `phasedb_findmatches` is used to set up and clean up database search jobs, and is run in the foreground. The database search is performed by the program `phase_dbsearch`. The setup mode of `phasedb_findmatches` sets up the input file for `phase_dbsearch`; the cleanup mode removes intermediate files generated during the search. The input file is named *jobname*`_dbsearch.inp`, and is described in detail in Section B.12 on page 189. The syntax for performing the three steps is as follows:

`$SCHRODINGER/utilities/phasedb_findmatches -setup` *jobname* `-db` *dbName*
`-hypo` *hypoID* `-mode` *runMode* `[-sub` *dbSubset*`] [-dtol` *deltaDist*`]`
`[-minSites` *minSites*`] [-matchFile` *matchFile*`] [-maxHits` *maxHits*`]`

`$SCHRODINGER/phase_dbsearch [`*job-options*`]` *jobname*

`$SCHRODINGER/utilities/phasedb_findmatches -cleanup` *jobname*

The options for `phasedb_findmatches` are given in Table 13.4. The job options for `phase_dbsearch` are given in Table 13.1. There is one additional job option for `phase_dbsearch`, `-BLOCK` *m*, which is used to process molecules in blocks of size *m*. This option forces memory cleanup every *m* molecules. The default is 5000 for standard searching and 1000 for flexible searching.

In addition to setting options to control the search and filter the hits, you can include certain files for the hypothesis in the current directory to perform filtering tasks. When these files are detected by `phasedb_findmatches`, the relevant file keyword is added to the database search input file. The files and the associated actions are described in Table 13.5.

To apply cutoffs on components of the fitness score, you must edit the database search input file and change the relevant keywords. For vector and volume scoring, a reference ligand must be provided. See Table B.12 for information on this file.

Table 13.4. Options for phasedb_findmatches

Option	Description
`-setup` *jobname*	Job name that will be used when launching `phase_dbsearch`. This determines the names of various input and output files.
`-db` *dbName*	Database name, including absolute path.
`-hypo` *hypoID*	Prefix for hypothesis files. At minimum, the files *hypoID*`.xyz` and *hypoID*`.def` must be present. To use a reference ligand, *hypoID*`.mae` and *hypoID*`.tab` must also be present.
`-mode` *runMode*	Database searching mode. Allowed values are `find+fetch`, `fetch`, and `flex`: `find+fetch`—Database of precomputed conformations is searched for geometric matches to to the hypothesis, all of which are written to a match file. Aligned conformations for the best matches are written to a hit file in Maestro format. `fetch`—Geometric matches from a previous `find+fetch` job are used to create the hit file. This allows different hit criteria to be applied without having to search the database again. If running in fetch mode, you must supply the name of the match file using the `-matchFile` option. `flex`—Conformations and pharmacophore sites are generated as the database is searched. These data are never written to disk, and no match file is produced, only a hit file.
`-sub` *dbSubset*	Database subset. The file *dbSubset*`_phase.inp` should reside in the current working directory, and it should contain `LIGAND_NAME` records for some subset of the database molecules. If this option is omitted, all molecules in the database are searched.
`-dtol` *deltaDist*	Intersite distance matching tolerance. Default: 2.0 Å.
`-minSites` *minSites*	Minimum number of hypothesis sites to match. Default: all sites must match.
`-matchFile` *matchFile*	Name of match file from a previous `find+fetch` job. Valid and required only when *runMode* is `fetch`.
`-maxHits` *maxHits*	Maximum number of hits to store. Default: 1000.

Table 13.5. Optional files and their associated actions

File	Action performed
hypoID`.dxyz`	Apply hypothesis-specific tolerances when matching.
hypoID`.mask`	Apply site mask when matching.
hypoID`.qsar`	Calculate predicted activities for the provided QSAR models.

Table 13.5. Optional files and their associated actions (Continued)

File	Action performed
hypoID.tol	Apply feature-based cutoffs
hypoID.xvol, *hypoID*.ev	Apply excluded volumes to filter matches

13.3.1 Examining Search Results

To view the hits in Maestro, import the file *jobname*-hits.mae. The following properties from the database search are added to the Project Table:

Ligand Name	Matched Ligand Sites	Volume Score
Conf Index	Align Score	Fitness
Num Sites Matched	Vector Score	Pred Activity(*n*)

The Pred Activity score is included only if a QSAR model was used. There will be as many predicted activity properties as there were PLS factors in the QSAR model, numbered according to the number of PLS factors in the model.

13.3.2 Searching Database Subsets

You can search database subsets by creating a subset with `phasedb_subset` (see Section 13.4 on page 150), and using the `-sub` option to `phasedb_findmatches`.

13.3.3 Applying Feature-Based Cutoffs

When you search for matches, the alignment cutoff factor specified by `-dtol` is used to eliminate molecules whose intersite distances are too large. This filtering procedure does not distinguish between different kinds of features. You might want to apply a tighter cutoff for a given feature type. These cutoffs can be specified by creating a feature-matching tolerances file and saving it as *hypoID*.tol in the current directory. The format of this file is described in Section B.9 on page 187. When you run `phasedb_findmatches`, the keyword `featureCutoffFile` is added to the database search input file.

Cutoffs based on feature types might be too restrictive in some cases. You can specify cutoffs for each feature in a hypothesis, by creating a hypothesis-specific tolerances file and saving it as *hypoID*.dxyz in the current directory. The format of this file is described in Section B.10 on page 188. When you run `phasedb_findmatches`, the keyword `featureCutoffFile` is added to the database search input file.

In the search, each match is checked to see whether any aligned site deviates from the hypothesis by more than the cutoff associated with that feature type or feature. The match is eliminated if a cutoff is exceeded.

13.3.4 Applying Excluded Volumes

Matches can also be filtered based on regions of space that should not be occupied by any part of the ligand, known as excluded volumes. You can create excluded volumes as part of the hypothesis generation—see Section 12.8 on page 132. To apply these excluded volumes to filter the matches, copy the excluded volume file *hypoID*`.xvol` or *hypoID*`.ev` to the current directory before running `phasedb_findmatches`. The `exclVolFile` keyword is added to the database search input file.

13.3.5 Searching for Partial Matches

You can search for partial matches to a hypothesis by setting the value of the `-minSites` option to a value less than the number of sites in the hypothesis. The hits that are returned can match any of the hypothesis features. If you want to require certain features to be included in the match, you can create a file that sets a mask for matching, and use it in the search by naming the file *hypoID*`.mask` and saving it in the current directory before you run `phasedb_findmatches`. The keyword `siteMaskFile` is then added to the database search input file. See Section B.11 on page 189 for information on site mask files.

For example, suppose you had the following hypothesis coordinate file (`.xyz` file):

```
2 A 7.714 -2.723 0.686
0 D 4.737 0.34 1.532
4 P 6.512 -0.62 3.508
8 R 0.852 0.512 1.119
7 R 2.102 1.913 -0.489
```

If you set partial matching with `minSites=3`, and you required the acceptor site and the positive site to be matched, the corresponding site mask file would be:

```
1
0
1
0
0
```

13.4 Creating Database Subsets: phasedb_subset

The `phasedb_subset` utility is used for creating and manipulating Phase database subset files. The command syntax is as follows:

```
$SCHRODINGER/utilities/phasedb_subset -db database {input-options}
    -out subset
```

The syntax of the input options is given in the following three usage scenarios. The descriptions of the options are given in Table 13.6.

Table 13.6. Options for phasedb_subset

Option	Description
`-db` *database*	Database name, including absolute path.
`-hits` *hitfile*	Hit file created from a previous search of the database.
`-out` *subOut*	Name of subset to create. The appropriate `LIGAND_NAME` records are written to the file *subOut*`_phase.inp`.
`-in1` *subset1*	First subset. `LIGAND_NAME` records must be stored in the file *subset1*`_phase.inp`.
`-in2` *subset2*	Second subset. `LIGAND_NAME` records must be stored in the file *sub2*`_phase.inp`.
`-logic` *operator*	Binary operator that specifies how to combine subsets. Allowed values of *operator* are `AND`, `OR`, and `NOT`: `AND` Records that are in both *sub1* and *sub2*. `OR` Records that are in either *sub1* or *sub2*. `NOT` Records that are in *sub1* but not *sub2*.
`-confs {true|false}`	Create subset of molecules with multiple conformers (`true`) or molecules with only a single conformer (`false`).
`-sites {true|false}`	Create subset of molecules with pharmacophore sites (`true`) or molecules without pharmacophore sites (`false`).

To create a subset from the structures in a hit file:

```
phasedb_subset -db database -hits hitfile -out subset
```

To create a subset from a logical operation on two existing subsets:

```
phasedb_subset -db database -in1 subset1 -logic operator -in2 subset2
    -out subset
```

To create a subset from a query of the database:

phasedb_subset -db *database* {-confs|-sites} {true|false} -out *subset*

This usage allows you to select molecules for which conformers have or have not been generated or for which sites have or have not been created.

13.5 Converting a Database: phasedb_convert

The utility `phasedb_convert` is a tool for reformatting Phase 3D databases. It supports forward conversion of databases from previous versions of Phase, and storage format changes to databases created using the current version of Phase. The command syntax is as follows:

$SCHRODINGER/utilities/phasedb_convert -source *dbSource* -target *dbTarget* [*job-options*] [*conversion-options*]

The job control options listed in Table 13.1 are supported. The `-JOB` *jobname* job option must be supplied, since this utility is run as a job. Conversion options are listed in Table 13.7.

Table 13.7. Conversion options for phasedb_convert

Option	Description
`-source` *dbSource*	The database containing the records to be converted. *dbSource* should be of the form *dbPath*/*dbName*. If the database was created using the Phase 1.0 GUI, *dbName* should always be `phasedb`. Required.
`-target` *dbTarget*	The database that will receive the converted records. *dbTarget* should also be of the form *dbPath*/*dbName*, but it cannot be the same as *dbSource*. Required.
`-new` *dbFileType*	Create a new target database with the specified storage format. *dbFileType* must be `mae` or `sd`, and it applies to all subsequent additions. This option is illegal if the target database already exists.
`-fd` *fdFile*	Use pharmacophore feature definitions in *fdFile* when creating a new target database. If this option is omitted, the default definitions in the Phase installation are used. This option is illegal if the target database already exists.
`-records` *recordFile*	File containing the source records to convert. *recordFile* should contain a series of lines of the form `LIGAND_NAME =` *sourceLigandName* If the entire database is to be converted, then *recordFile* may be *dbSource*`_master_phase.inp` or *dbSource*`_phase.inp`, whichever one contains the full set of records in the database. Required unless `-restart` is used.

Table 13.7. Conversion options for phasedb_convert

Option	Description
`-restart`	Rerun the conversion. Use only if a previous attempt failed to successfully convert all records.
`-final`	Final attempt to convert. Abandon any records that fail to convert.
`-delete`	Delete source files as the associated records are successfully converted. If this option is omitted, the source database will not be modified. Even if `-delete` is used, some files and directories will remain in the source database. These can be removed by hand once all source records are converted or abandoned.

13.6 Running on Multiple Processors

Site creation jobs and database search jobs can be split across multiple processors on an appropriately configured cluster. The basic requirements for database creation and searching are as follows:

- The database must be located in a directory that is uniformly accessible to all nodes of the cluster on which jobs will be run.
- If the file system on which the database is stored is only accessible to the cluster, you must be logged into the manager node of the cluster to start jobs.
- In the `$SCHRODINGER/schrodinger.hosts` file, each parallel queue that is used for database jobs should have a `tmpdir` entry with a path that is accessible to all nodes. See the *Installation Guide* for details.

When you start a `phase_dbcreate` or `phase_dbsearch` job, the `-HOST` option should specify the processors to use. If the processors are on a single host, you can add the number of processors after the host name and a colon—for example, `-HOST cluster:4`. One subjob is started on each processor. The molecules in the database are distributed to the subjobs, one at a time, until the list is exhausted. For this example, the first subjob would process molecules 1, 5, 9, and so on; the second subjob would process molecule 2, 6, 10, and so on. This scheme provides near optimal load balancing as it is very unlikely that any one processor would have to process a disproportionate number of expensive molecules.

Searching Files for Matches from the Command Line

In addition to searching a previously prepared database for matches, Phase provides two programs for searching files for matches. The first, `phase_fileSearch`, can be run only on a single processor. The second, `phase_gridSearch`, can be run on multiple processors, including a grid of Linux hosts under the management of a United Deviced GridMP grid server. For information on installing the software for use on this kind of grid, see page 46 of the *Installation Guide*.

The same requirements on the structures apply to searching a file as for adding structures to a database: the structures must be all-atom, 3D structures. If the structures do not meet these requirements, you should convert them using LigPrep, for example (see the *LigPrep User Manual*).

14.1 Searching Files with phase_fileSearch

If you have a relatively small number of structures to search for matches, you can run `phase_fileSearch`. This program only runs on a single processor, so is not suited for large numbers of structures. If you want to distribute a large set of structures across multiple processors, consider using `phase_gridSearch` or creating a database and using `phase_dbsearch`.

Before you use `phase_fileSearch`, you must set up the input file for the search. You can do this with the utility `pharm_align_mol`. See Section 12.9.2 on page 136 for a description of this utility, and see page 111 for a summary of the syntax of using `pharm_align_mol` with `phase_fileSearch`.

Syntax

`phase_fileSearch [`*job-options*`]` *jobname*

The file *jobname*`_fileSearch.inp` must contain the keyword=value pairs required to run this job. This file can be set up with the utility `pharm_align_mol`. The standard job control options listed in Table 13.1 can be used with this program.

14.2 Searching Files with phase_gridSearch

For large sets of structures that are stored in SD format, you can use `phase_gridSearch` to search for matches. This program can be run on multiple CPUs and on a Linux grid under GridMP. Running on a grid requires that you install Phase on each host on the grid, and set up an entry in `schrodinger.hosts` that directs jobs to the grid. For information on installing the software for use on this kind of grid, see page 46 of the *Installation Guide*. If you choose only a single processor for this job, the job runs on the local host regardless of which host you selected. This program is not intended for single-processor use, so you should always ensure that multiple processors are specified with the `-HOST` option.

When you run `phase_gridSearch`, the search input file is prepared automatically using the options that you specify. However, you must also prepare a file that contains a list of files to be searched, one file name per line (even if you only want to search one file). The file names should include the absolute path to the file. All files should be either standard SD files or compressed SD files. You cannot have both standard and compressed files in the same run.

The structures that are searched are distributed across the available processors by adding structures one at a time to temporary files, one file per processor. When a predetermined number of structures has been added to a temporary file, the file is closed and compressed with `gzip`, and a new temporary file is opened for that procesor. If you are using a large number of processors, you may want to set the number of structures per temporary file to a small value to ensure that these files do not take up too much disk space. The distribution process is run on the local host and the temporary files are stored in the current working directory, so you should ensure that you have enough disk space in this directory to store all the temporary files.

Syntax

`phase_gridSearch -structFileList` *fileName* `-hypoID` *hypoID* `-maxHits` *maxHits*
`[`*options*`] [`*job-options*`]` *jobname*

The arguments and options are described in Table 14.1. These options correspond to keywords in the database search input file, which is described in detail in Section B.12 on page 189. The standard job control options listed in Table 13.1 can be used with this program. One additional job option is listed at the end of Table 14.1.

Table 14.1. Options for phase_gridSearch

Option	Description
`-structFileList` *fileName*	Name of the file containing the list of SD files to search. The SD files may be all in standard form, or all compressed via gzip, but mixing of standard and compressed files is not permitted. Compressed file names should have a `.gz` extension.
`-hypoID` *hypoID*	Prefix used to name all hypothesis files. The files *hypoID*`.def` and *hypoID*`.xyz` must always be present. Other files are optional.
`-maxHits` *maxHits*	Maximum total number of hits that will be returned in the file *jobname*`-hits.mae`. If the maximum limit is reached, only the best hits are kept.
`-flexSearchMethod` *method*	Conformational sampling method. Allowed values are `rapid` and `thorough`. Default: `rapid`.
`-flexMaxConfs` *maxConfs*	Maximum number of conformations/molecule to generate. If zero, no conformations will be generated, and the supplied structures will be searched directly. Default: 100.
`-flexMaxRelEnergy` *energy*	Conformational energy window in KJ/mol. Default: 41.84 kJ/mol (10 Kcal/mol).
`-deltaDist` *deltaDist*	Intersite distance matching tolerance in angstroms. Default: 2.0.
`-minSites` *minSites*	The minimum number of hypothesis sites that must be matched. Must be 3 or greater. The default is to match all sites.
`-alignWeight` *alignWeight*	Alignment score weight. Must be non-negative. Default: 1.0.
`-alignCutoff` *alignCutoff*	Alignment score cutoff. Must be greater than zero. Default: 1.2.
`-alignPenalty` *alignPenalty*	Partial matching alignment penalty. Must be non-negative. Default: 1.2.
`-vectorWeight` *vectorWeight*	Vector score weight. Must be non-negative. Relevant only when the hypothesis has a reference ligand. Default: 1.0.
`-vectorCutoff` *vectorCutoff*	Vector score cutoff. Must be on the interval [-1, 1]. Relevant only when the hypothesis has a reference ligand. Default: -1.0.
`-volumeWeight` *volumeWeight*	Volume score weight. Must be non-negative. Relevant only when the hypothesis has a reference ligand. Default: 1.0.
`-volumeCutoff` *volumeCutoff*	Volume score cutoff. Must be on the interval [0, 1]. Relevant only when the hypothesis has a reference ligand. Default: 0.0.

Table 14.1. Options for phase_gridSearch (Continued)

Option	Description
`-useRefLigand {true|false}`	Use reference ligand information. If this option is set to `true`, the files *hypoID*`.mae` and *hypoID*`.tab` must be present, or the job will fail. Default: use a reference ligand if the files *hypoID*`.mae` and *hypoID*`.tab` are present.
`-useSiteMask {true|false}`	Apply a site mask in partial matching (i.e., requiring certain sites to match). If this option is set to `true`, the file *hypoID*`.mask` must be present, or the job will fail. Default: apply a site mask if the file *hypoID*`.mask` is present.
`-useFeatureCutoffs {true|false}`	Apply feature-based matching tolerances. If this option is set to `true`, the file *hypoID*`.tol` must be present, or the job will fail. Default: apply these tolerances if the file *hypoID*`.tol` is present.
`-useDeltaHypo {true|false}`	Apply hypothesis-specific matching tolerances. If this option is set to `true`, the file *hypoID*`.dxyz` must be present, or the job will fail. Default: apply these tolerances if the file *hypoID*`.dxyz` is present.
`-useQSARModel {true|false}`	Apply a QSAR model to the hits. If this option is set to `true`, the file *hypoID*`.qsar` must be present, or the job will fail. Default: apply a QSAR model if the file *hypoID*`.qsar` is present.
`-useExclVol {true|false}`	Apply excluded volumes to filter the hits. If this option is set to `true`, the file *hypoID*`.xvol` must be present, or the job will fail. Default: apply excluded volumes if the file *hypoID*`.xvol` is present.
`-maxHitsPerMol` *max-PerMol*	Maximum number of hits per molecule. Default: 1.
`-MAXSTRUCT` *n*	Maximum number of structures written to each temporary input file before it is compressed. The default is 1000. Use a smaller value to reduce the amount of uncompressed data on disk at any given time.

Detecting Multiple Binding Modes

Some receptor sites permit ligand binding in distinct binding modes, rather than a single binding mode. Phase includes an automated, scripted procedure to help you identify subsets of ligands that bind in distinct modes. This is done by applying a clustering algorithm to a large number and variety of common pharmacophore hypotheses derived from the ligands of interest. Clustering is based on a bit string similarity metric, in which each ligand is assigned a bit, and the bits set for a given hypothesis correspond to the ligands that contain (i.e. match) that hypothesis. Thus the similarity between hypotheses is measured to be high if they have many bits (i.e., many ligands) in common.

As an example, suppose you have 30 ligands and this procedure has been applied to identify two highly distinct clusters. Suppose further that hypotheses A and B are representatives from each cluster, and that 13 of the ligands match hypothesis A, while the remaining 17 match hypothesis B. In this case, perfect separation is achieved, and hypotheses A and B provide a model to explain the multiple binding modes.

In practice, the separation may not be perfect, and a given cluster may contain very different families of pharmacophores, so selecting a representative may not always be straightforward. Because of this, the critical goal is identifying the most probable subsets of ligands, rather than specific pharmacophore models. Once the ligand subsets are known, each can be pursued independently to fine-tune the associated pharmacophore models.

To facilitate identification of subsets, visualization tools are employed. A 2-dimensional “heat map” is provided to illustrate the association of hypotheses, ligands and clusters. If large, distinct clusters are apparent, then evidence exists for multiple binding modes and the appropriate ligand subsets can be pursued.

When generating common pharmacophore models, there are a few parameters that must be assigned using a reasonable amount of discretion. For example, minimum and maximum limits must be set on the number of sites in the pharmacophores to be generated. The software supports a range of 3 to 7 sites, but in most cases it is sufficient to consider only 4- and 5-point pharmacophores.

The minimum number of ligands that must match each pharmacophore must also be chosen, and hence the minimum number of ligands that is needed to establish a binding mode. In theory, requiring a minimum of 2 ligands per binding mode would provide exhaustive identification of potential binding modes, but only at tremendous computational expense. It is strongly recommended that this value be no smaller than it needs to be. For example, if you have 30 ligands, and you suspect that they bind in 2 equally probable modes, then it is reasonable to require that at least 10 ligands match each pharmacophore.

Once common pharmacophores are generated, the usual scoring procedure is performed to rank hypotheses and eliminate those that provide less satisfactory alignments. If you wish to consider pharmacophore models with different numbers of sites (e.g., 4-point and 5-point pharmacophores), then separate Phase runs should be created within the same project, branched at the common pharmacophore step. Note that the exact same ligands should be used in all runs, and the same minimum number of matching ligands should be required.

After generating scoring results for one or more runs, you can run the Phase Cluster Visualization Python script, `clusterVis.py`. This script opens a panel that allows you to select one or more runs, the number of clusters, and clustering options. To run this script you must first install it from `$SCHRODINGER/python-v`*version*`/scripts/maestro`, then choose Phase Cluster Visualization Tool from the Scripts menu.

The number of clusters should be equal to the number of binding modes that are believed to exist. This does not affect how the clustering is performed, it merely changes the location of the vertical yellow lines that are drawn on the heat map, which suggest the most probable splitting points between ligand clusters.

To perform a clustering calculation:

1. Select the runs in the Project Data section.
2. Select options from the option menus in the Clustering Options section.

   These option menus are described below.
3. Click Calculate.

Option menus

Metric menu

Select the method used to compute distances between bit strings representations.

- Euclidean—Sum of the squares of the differences in the bit string values. This is just the number of positions at which two bit strings differ.
- Tanimoto—Tanimoto metric, 1 – N(common)/N(total), where N(common) is the number of ‘on’ bits shared by the two strings (bit intersection), and N(total) is the total number ‘on’ bits in either string (bit union).
- Cosine—Cosine of the angle between the two bit string vectors.

Linkage menu

Select the method used to compute distances between clusters.

- Single—The distance between clusters is the smallest distance between any pair of objects (one object from each cluster). This option produces diffuse, elongated clusters.
- Average—The distance between clusters is the average distance between all pairs of objects in the two clusters.
- Complete—The distance between clusters is the largest distance between any pair of objects (one object from each cluster). This option produces compact, spherical clusters.

Getting Help

Schrödinger software is distributed with documentation in PDF format. If the documentation is not installed in `$SCHRODINGER/docs` on a computer that you have access to, you should install it or ask your system administrator to install it.

For help installing and setting up licenses for Schrödinger software and installing documentation, see the *Installation Guide*. For information on running jobs, see the *Job Control Guide*.

Maestro has automatic, context-sensitive help (Auto-Help and Balloon Help, or tooltips), and an online help system. To get help, follow the steps below.

- Check the Auto-Help text box, which is located at the foot of the main window. If help is available for the task you are performing, it is automatically displayed there. Auto-Help contains a single line of information. For more detailed information, use the online help.
- If you want information about a GUI element, such as a button or option, there may be Balloon Help for the item. Pause the cursor over the element. If the Balloon Help does not appear, check that Show Balloon Help is selected in the Help menu of the main window. If there is Balloon Help for the element, it appears within a few seconds.
- For information about a panel or the tab that is displayed in a panel, click the Help button in the panel. The Help panel is opened and a relevant help topic is displayed.
- For other information in the online help, open the Help panel and locate the topic by searching or by category. You can open the Help panel by choosing Help from the Help menu on the main menu bar or by pressing CTRL+H.

  To view a list of all available Phase–related help topics, click the Categories tab, then from the Categories menu, choose Phase. Double-click a topic title to view the topic.

If you do not find the information you need in the Maestro help system, check the following sources:

- *Maestro User Manual*, for detailed information on using Maestro
- *Maestro Command Reference Manual* for information on Maestro commands
- *Phase Quick Start Guide*, for a tutorial guide to using Phase
- Frequently Asked Questions pages, at https://www.schrodinger.com/Phase_FAQ.html

The manuals are also available in PDF format from the Schrödinger Support Center. Information on additions and corrections to the manuals is available from this web page.

If you have questions that are not answered from any of the above sources, contact Schrödinger using the information below.

E-mail: help@schrodinger.com
USPS: 101 SW Main Street, Suite 1300, Portland, OR 97204
Phone: (503) 299-1150
Fax: (503) 299-4532
WWW: http://www.schrodinger.com
FTP: ftp://ftp.schrodinger.com

Generally, e-mail correspondence is best because you can send machine output, if necessary. When sending e-mail messages, please include the following information, most of which can be obtained by entering `$SCHRODINGER/machid` at a command prompt:

- All relevant user input and machine output
- Phase purchaser (company, research institution, or individual)
- Primary Phase user
- Computer platform type
- Operating system with version number
- Phase version number
- Maestro version number
- mmshare version number

Phase QSAR Models

A.1 The Phase QSAR Methods

Phase QSAR models are developed from a series of molecules, of varying activity, that have all been aligned to a common pharmacophore hypothesis that is associated with a single reference ligand. QSAR models may be atom-based or pharmacophore-based: the former takes all atoms into account; the latter uses the pharmacophore sites that can be matched to the hypothesis.

A rectangular grid is defined to encompass the space occupied by a training set of aligned molecules. This grid divides space into uniformly-sized cubes, typically one angstrom on each side. In atom-based models, the grid is populated by van der Waals spheres, with radii that depend on the atom type. In pharmacophore-based models, the grid is populated by the pharmacophore sites that match the hypothesis, with each site represented by a sphere with a user-definable radius.

A given atom or pharmacophore site will occupy the space of one or more cubes in the grid. Occupation of a cube is deemed to occur if the center of that cube falls within the radius of the atom or site. A given cube may be occupied by more than one atom or site, and that occupation may come from the same molecule or from different molecules.

Each occupied cube gives rise to one or more volume bits. A volume bit is allocated for each different class of atom or site that occupies a cube.

In pharmacophore-based models, sites are assigned to classes that are determined by the feature definitions used to create the hypothesis (e.g., A, D, H, N, P, R). In atom-based models, there are 6 distinct atom classes that have some correspondence or similarity with pharmacophore feature types, but atom classes are assigned using fixed internal rules, not the hypothesis feature definitions:

- D – Hydrogen bond donor (hydrogens bonded to N, O, P, S)
- H – Hydrophobic/non-polar (C, H–C, Cl, Br, F, I)
- N – Negative ionic (formal negative charge)
- P – Positive ionic (formal positive charge)
- W – Electron-withdrawing (N, O)
- X – Miscellaneous (all other types of atoms)

So, if a particular cube is occupied by a “D” from molecule 1, an “H” from molecule 5, and a “P” from molecule 9, that cube would be allocated three volume bits. If a cube is never occupied by any molecule in the training set, no volume bits would be allocated. Hence, each volume bit must be set by at least one molecule in the training set.

The pool of volume bits provides a means of characterizing the molecules. In atom-based models, the pattern of volume bits that are set by a molecule encodes the size, shape, and chemical characteristics of that molecule. In pharmacophore-based models, the pattern of set bits determines which subset of critical pharmacophore features that molecule contains, and the positions of those features in relation to other molecules.

If a binary scheme (0/1) is used to denote which bits are set by each molecule, a table of bit values may be assembled:

$molecule_1$	0	1	1	0	0	1	1	0	0	1	0	1	0	.	.	.
$molecule_2$	0	1	1	0	0	0	1	0	0	1	0	1	0	.	.	.
$molecule_3$	0	0	0	1	0	1	0	0	1	0	1	0	1	.	.	.
	.															
	.															
	.															

For atom-based models, there are usually several hundred or more volume bits for each series of aligned molecules. For pharmacophore-based models, that number is much smaller, usually only a few dozen. The number of bits increases as the grid spacing becomes finer, and, in the case of atom-based models, as the molecules become larger.

To generate a QSAR model, the 0/1 bit values are treated as independent variables in partial least-squares (PLS) regression analysis. This involves finding a linear least-squares relationship between the activity data and a special set of orthogonal factors that are linear combinations of the bit value variables.

More precisely, if there are n molecules in the training set and v volume bits, let the $n \times v$ matrix $\mathbf{X}$ represent the table of volume bits, and let the $n \times 1$ vector $\mathbf{y}$ represent the activity values for the training set molecules. Creation of the PLS regression model proceeds as follows:

Center each column of $\mathbf{X}$:

for $i = 1,\ldots,v$

$$\mu_i^x = \frac{1}{n}\sum_{k=1}^{n} X_{k,i}$$

for $k=1,\ldots,n$

$$X_{k,i} \rightarrow X_{k,i} - \mu_i^x$$

next k

next i

Center **y**:

$$\mu^y = \frac{1}{n}\sum_{k=1}^{n} y_k$$

for k=1,…,n

$$y_k \rightarrow y_k - \mu^y$$

next k

Determine PLS factors and regression coefficients for up to m PLS factors ($m \le v$):

$\mathbf{X}_1 = \mathbf{X}$

for i = 1,…,m

Compute the vector of weights that define PLS factor i:

$$\mathbf{w}_i = \mathbf{X}_i^{\mathrm{T}}\mathbf{y} \big/ \left|\mathbf{X}_i^{\mathrm{T}}\mathbf{y}\right|, \qquad \mathbf{w}_i \in \mathbf{R}^{v \times 1}$$

Project the rows of $\mathbf{X}_i$ onto factor i:

$$\mathbf{t}_i = \mathbf{X}_i \mathbf{w}_i, \qquad \mathbf{t}_i \in \mathbf{R}^{n \times 1}$$

Project $\mathbf{t}_i$ onto each column of $\mathbf{X}_i$:

$$\mathbf{p}_i = \mathbf{X}_i^{\mathrm{T}}\mathbf{t}_i \big/ \left|\mathbf{t}_i^{\mathrm{T}}\mathbf{t}_i\right|, \qquad \mathbf{p}_i \in \mathbf{R}^{v \times 1}$$

Compute the ith PLS regression coefficient by projecting $\mathbf{t}_i$ onto **y**:

$$\mathbf{b}_i = \mathbf{t}_i^{\mathrm{T}}\mathbf{y} \big/ \left|\mathbf{t}_i^{\mathrm{T}}\mathbf{t}_i\right|, \qquad \mathbf{b}_i \in \mathbf{R}^{m \times 1}$$

Orthogonalize $\mathbf{X}_i$ with respect to PLS factor i:

$$\mathbf{X}_{i+1} = \mathbf{X}_i - \mathbf{t}_i\,\mathbf{p}_i^{\mathrm{T}}$$

next i

For a regression with m PLS factors, the fitted activities are then given by:

$$\hat{\mathbf{y}} = \mu^y + \sum_{i=1}^{m} b_i\,\mathbf{t}_i$$

To apply the m-factor PLS model to a new set of n_T ligands with bit value matrix $\tilde{\mathbf{X}}$, the regression coefficients $\mathbf{b}$ must first be translated back to the space of the original $\mathbf{X}$ variables:

Define

$$\mathbf{W} \equiv [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m], \qquad \mathbf{W} \in \mathbf{R}^{v \times m}$$

$$\mathbf{P} \equiv [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_m], \qquad \mathbf{P} \in \mathbf{R}^{v \times m}$$

Then

$$\mathbf{b}^x \equiv \mathbf{W}(\mathbf{P}^{\mathrm{T}}\mathbf{W})^{-1}\mathbf{b} \qquad \mathbf{b}^x \in \mathbf{R}^{v \times 1}$$

The coefficients $\mathbf{b}^x$ may then be used to predict activities for the new ligands:

$$\hat{y}_k = \mu^y + \sum_{i=1}^{m} (\tilde{X}_{k,i} - \mu_i^x)\, b_i^x \qquad k = 1, \ldots, n_T$$

A.2 Phase QSAR Statistics

This section defines the various statistical measures that are used in Phase QSAR models.

A.2.1 Training Set and Model

Statistical quantities describing the training set and the QSAR model are defined below.

m	number of PLS factors in the model
n	number of molecules in the training set
$df_1 = m + 1$	degrees of freedom in model
$df_2 = n - m - 2$	degrees of freedom in data
y_i	observed activity for training set molecule i
$\hat{y}_i$	predicted activity for training set molecule i
$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$	mean observed activity

$\sigma_y^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2$ variance in observed activities

$sse = \sum_{i=1}^{n}(\hat{y}_i - y_i)^2$ sum of squared errors

$\sigma_{\mathrm{err}}^2 = \frac{sse}{n}$ variance in errors

$ssy = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$

$SD = \sqrt{sse/df_2}$ standard deviation of regression

$R^2 = 1 - \frac{\sigma_{\mathrm{err}}^2}{\sigma_y^2}$ R-squared; coefficient of determination

$F = \frac{ssy/df_1}{sse/df_2}$ F statistic; overall significance of model

$P = B(df_1, df_2, \frac{df_2}{df_2 + Fdf_1})$ statistical significance; probability that correlation could occur by chance. The beta function $B(a, b, x)$ is defined by

$$B(a, b, x) = \int_0^x t^{a-1}(1-t)^{b-1}dt$$

Note that R^2 can never be negative, because the regression coefficients are optimized to minimize *sse*. The worst-case scenario is when the independent variables have absolutely no statistical relationship with activity. Under those circumstances, the regression coefficients will all be zero, and the model will contain only an intercept parameter, the value of which will be $\bar{y}$. Thus every predicted activity will be $\bar{y}$, and σ_{err}^2 will be equal to σ_y, yielding $R^2 = 0$.

A.2.2 Test Set Predictions

Statistical quantities describing the test set predictions are described below.

T	The test set of molecules
n_T	number of molecules in T
y_j	observed activity for molecule $j \in T$
$\hat{y}_j$	predicted activity for molecule $j \in T$

$$RMSE = \sqrt{\frac{1}{n_T} \sum_{j \in T} (\hat{y}_j - y_j)^2}$$ root-mean-squared error

$$Q^2 = R^2(T)$$ Q-squared

$$r = \frac{\sum_{j \in T} (y_j - \bar{y}_T)(y_j - \bar{\hat{y}}_T)}{\sqrt{\sum_{j \in T} (y_j - \bar{y}_T)^2 (y_j - \bar{\hat{y}}_T)^2}}$$ Pearson *r* value, Pearson correlation coefficient

The formulas for R^2 and Q^2 are equivalent, with the only difference being that Q^2 is computed using the observed and predicted activities for the test set. However, Q^2 can take on negative values. This happens whenever the variance in the test set errors is larger than the variance in the observed test set activities. Often, the test set does not have as large a range of activity values as the training set (so the variance in *y* is smaller), and the errors for the test set tend to be larger than those for the training set (so the variance in the errors is larger), It is therefore not uncommon to see negative Q^2 values from time to time.

Because all values are shifted by the sample means, the Pearson correlation coefficient is insensitive to systematic errors in the predictions, whereas Q^2 is not. So if the rank order of the activity predictions is basically correct, but there's a significant constant shift in the values compared to the observed activities, the Pearson correlation coefficient may still be quite high, even if Q^2 is small or negative.

Phase Input Files

In addition to structure files, Phase uses a variety of data input files. These files are described in the following sections. Normally you would not need to edit most of these files, as they are set up using the command-line utilities or from Maestro. However, some of the files used when searching for matches must be created by hand.

In this appendix, references to *utilities* are to programs or scripts in the `$SCHRODINGER/utilities` directory.

B.1 Master Data File

This file stores various pieces of ligand data that are used throughout the pharmacophore model development workflow. It is named `MasterData.tab`, and can be updated by the utility `pharm_data` or by hand. At the top of the file is a description of the data it contains, and a number of rules regarding how the data may be modified. The body of the file contains first a ligand property name block, followed by a set of ligand blocks, one per ligand in the project.

If you make any changes to `MasterData.tab`, whether by hand or through operations supported by the `pharm_data` utility, you must run `pharm_data` with the `-commit` option to update the Maestro files in the `ligands` subdirectory. If you do not update the Maestro files, various Phase modules will not be using the modified values because they read the property data directly from the Maestro files.

If you have completed any forward steps in the project workflow, the results generated in those steps may be invalidated by changes you make to `MasterData.tab`. When you attempt to commit the changes, you will be supplied with a list of forward files that will be invalidated, and you will be given a chance to abort the commit operation. If you choose to abort, you can rerun `pharm_data` with the `-restore` flag to revert to the previous version of the file, which is stored in `MasterData.backup`.

The ligand property name block keywords are described in Table B.1. These properties are set by `pharm_project` and must not be altered by hand. This block defines the names of certain properties that are relevant to the pharmacophore model development. The Maestro files in the `ligands` subdirectory contain these named properties.

Table B.1. Ligand property block keywords in the Master Data file.

Keyword	Description
`LIGAND_NAME_PROPERTY`	Name of Maestro property that defines the ligand name. Set automatically by `pharm_project` to `s_phase_Ligand_Name`.
`PHARM_SET_PROPERTY`	Name of Maestro property that defines the pharm set membership. Set automatically by `pharm_project` to `s_phase_Pharm_Set`.
`QSAR_SET_PROPERTY`	Name of Maestro property that defines the QSAR set membership. Set automatically by `pharm_project` to `s_phase_QSAR_Set`.
`ACT_PROPERTY`	Name of the property that stores the ligand activity values. Set by the `-act` option of `pharm_project`.
`1D_PROPERTY`	Name of Maestro property that defines the `1D_VALUE` used to control which ligands can give rise to hypotheses. Set automatically by `pharm_project` to `r_phase_Ligand_1D_Property`.
`CONF_PROPERTY`	Name of Maestro property that defines the conformation-dependent quantity used in scoring. This is normally the relative conformational energy. Set by the `-conf` option of `pharm_project`.

The keywords that define the ligand data values in the ligand blocks are described in Table B.2. These data values are stored in the ligand Maestro files in addition to the master data file. Only the conformation-independent properties are kept in these blocks: the conformation-dependent property defined by `CONF_PROPERTY` is not stored here, but only in the ligand Maestro file.

Table B.2. Ligand block keywords in the Master Data file.

Keyword	Description
`LIGAND_NAME`	Ligand name, in the project. This is not the same as the title or the Maestro entry name, but is a unique identifier used in the pharmacophore model project. Do not modify.
`TITLE`	Title of the ligand. Taken from the Title property in the Maestro file. Do not modify.
`PHARM_SET`	Pharm set membership of the ligand. Can be modified by hand or using the `-active` and `-inactive` options of the `pharm_data` utility. Allowed values are `active`, `inactive`, and `none`. `active` Ligand is used to identify common pharmacophores and to score hypotheses. There must be at least two ligands with `PHARM_SET = active` `inactive` Ligand is used to measure the degree to which hypotheses discriminate actives from inactives by inactive scoring. `none` Ligand is not used in pharmacophore model development, but may be used in QSAR model development.

Table B.2. Ligand block keywords in the Master Data file.

Keyword	Description
QSAR_SET	QSAR set membership of the ligand. Affected by the `-train`, `-rand`, and `-pharm_set` options of the `pharm_data` utility. Can be modified by hand. Allowed values are `train`, `test`, and `none`. `train` Ligand is used to develop QSAR models. You should use at least five training set ligands for each PLS factor. `test` Ligand is used to test QSAR models. `none` QSAR models are not applied to this ligand.
ACTIVITY	Ligand activity. Affected by the `-log`, `-exp`, and `-multiply` options of the `pharm_data` utility. Can be modified by hand. Values should increase as potency increases, as for example in –log K_i or –log IC50. If activity is unknown, the value should be set to `missing`.
1D_VALUE	A conformationally independent numerical property that may be used during hypothesis scoring to influence or control the selection of reference ligands. This property is added to the actives score when the `-prop` option is used with `pharm_score_actives`. Must be set by hand.

An example of the top of a Phase Master Data file is given below. This excerpt includes the header, the ligand property name block, and a few ligand blocks.

```
################################################################################
#                                                                              #
# Phase Master Data File                                                       #
#                                                                              #
# You may change PHARM_SET, QSAR_SET, ACTIVITY and 1D_VALUE.  To propagate     #
# these changes to the project, use 'pharm_data -commit'.  To revert to the    #
# most recently committed version of the file, use 'pharm_data -restore'.      #
#                                                                              #
# PHARM_SET: Allowed values are "active", "inactive" and "none".              #
#            active   - Used to identify common pharmacophores and to score    #
#                       hypotheses.  There must be at least two ligands        #
#                       with PHARM_SET = active.                               #
#            inactive - Used to measure the degree to which hypotheses         #
#                       discriminate actives from inactives.                   #
#            none     - Not used in pharmacophore model development.           #
#                                                                              #
# QSAR_SET:  Allowed values are "train", "test" and "none".                    #
#            train - Used to develop QSAR models.  Recommend at least five     #
#                    training set ligands for each PLS factor.                 #
#            test  - Used to test QSAR models.                                 #
#            none  - QSAR models not applied to these ligands.                 #
#                                                                              #
# ACTIVITY:  Ligand activity.  Values should increase as potency increases,    #
#            for example, -logKi or -logIC50.  If activity is unknown, the     #
#            value should be "missing".                                        #
```

```
#                                                                              #
# 1D_VALUE:  A conformationally-independent numerical property that may be     #
#            used during hypothesis scoring to influence or control the        #
#            selection of reference ligands.                                   #
#                                                                              #
################################################################################
LIGAND_NAME_PROPERTY = s_phase_Ligand_Name
PHARM_SET_PROPERTY = s_phase_Pharm_Set
QSAR_SET_PROPERTY = s_phase_QSAR_Set
ACT_PROPERTY = r_phase_Ligand_Activity
1D_PROPERTY = r_phase_Ligand_1D_Property
CONF_PROPERTY = r_mmod_Relative_Potential_Energy-MMFF94s
################################################################################
LIGAND_NAME = mol_1
TITLE = "endo-1"
PHARM_SET = active
QSAR_SET = train
ACTIVITY = 5.509
1D_VALUE = 0.0
################################################################################
LIGAND_NAME = mol_2
TITLE = "endo-2"
PHARM_SET = active
QSAR_SET = train
ACTIVITY = 5.456
1D_VALUE = 0.0
```

B.2 Phase Main Input File

The Phase main input file contains information that is used across the entire Develop Pharmacophore Model workflow. It contains sections relevant to all programs in the workflow and sections that are used only by specific programs. This file is created by the model development utilities. If you edit this file, the order of ligands must remain unchanged for an entire run. When you run Phase from Maestro, Maestro creates and updates a main input file for each run.

Each line of the Phase main input file contains a keyword-value pair separated by an equals sign ("="), as follows:

keyword=value

Extra spaces are ignored, but blank lines are not permitted. Keywords can have string, integer, or real types. These types are enforced.

Keywords for the job name, file names, and directories are given in Table B.3. In the tables, the same text is used for the value as for the keyword, but it is set in a different font: for example, *ligand-name* represents the value of the keyword `LIGAND_NAME`.

Optional keywords for the Find Common Pharmacophores step are described in Table B.4. Optional keywords for the Score Hypotheses step are given in Table B.5. Each Score Hypotheses keyword has a default value, so it is not necessary to include any of them in the input file.

Changes to this file should be made with the utility programs described in Chapter 12.

Table B.3. Name and directory keywords.

Keyword	Description
`JOB_NAME`	Name given to each Phase job run with this input file. This name is used as a base for input and output files for the run. The name must match the name of the input file, *jobname*`_phase.inp`.
`LIGAND_DIR`	Directory where all ligand-related files are stored (the *ligands directory*). Should be a relative path. The ligand input files are multi-conformer Maestro files named *ligand-name*`.mae`. The output files include the conformer coordinate files, named *ligand-name*`_xyz.phc`, and the ligand sites files, named *ligand-name*`_sites.phs`. Optional keyword; the default is `ligands`.
`LIGAND_NAME`	Name of a ligand. This name is used to construct file names for ligand-related files. The ligand structure is contained in the file *ligand-name*`.mae`. For example, if the input file contains the line `LIGAND_NAME=aspirin`, there should be a Maestro file named `aspirin.mae` in the ligands directory. The input file should contain multiple lines of this kind, one for each ligand in the set. The order of these lines should *not* be changed during a run. For pharmacophore model development, the set should include only the active ligands on which the model is to be based. Do not change by hand.
`BOXES_DIR`	Name of directory where box files are stored. Box files are generated during the Find Common Pharmacophore step. These files contain data that is used as input for the scoring step. This keyword is used for both the Find Common Pharmacophore and Score Hypothesis steps. Optional keyword; the default is `boxes`.
`RESULT_DIR`	Name of the directory where results of the Score Hypothesis step are stored. Optional keyword; the default is `result`.

Table B.4. Optional Keywords for the Find Common Pharmacophores step.

Keyword	Description
NUM_SITES	Total number of sites in pharmacophore hypothesis (integer). Default is 5.
MIN_INTERSITE_DIST	Minimum distance between pharmacophore sites in angstroms (real). May be used to reject pharmacophores that contain, for example, an acceptor site and a donor site from the same oxygen. Default is 2.0 Å.
MAXIMUM_DEPTH	Number of times each side of the "box" is divided (integer). Default is 5.
INITIAL_BOX_SIZE	Initial box size in angstroms (real). This option should not appear in the Phase main input file. Set automatically using values for FINAL_BOX_SIZE and MAXIMUM_DEPTH as described below.
FINAL_BOX_SIZE	Final box size in angstroms (real). Default is 2.0 Å.
MIN_NUM_LIGANDS_PER_BOX	Minimum number of ligands that must be matched (integer).
MIN_MAX_SITES	Minimum and maximum number of sites for a feature type. Value is a string that contains 3 integers separated by commas with no spaces: *n1*,*n2*,*n3*. The first integer (*n1*) is the numerical code for the site type (see VARIANT_NAMES, below). The second integer (*n2*) is the minimum feature frequency and third integer (*n3*) is the maximum feature frequency. The maximum value of *n3* is 4. By default, the values of *n2* and *n3* are set to 0 and 4 for the standard features (A, D, H, N, P, and R), and to 0 and 0 for the custom features. If you change the defaults, the input file should contain multiple lines of this kind, one for each feature type.
VARIANT_NAMES	Comma-separated list of variants for which common pharmacophores are to be identified. These names are used to construct file names for variant-related files. Multiple lines of this kind can be used to specify the variants. By default, all variants are used. Each variant is a string of single-digit numbers in ascending order. The numbers encode the feature types, as follows: 0 Hydrogen-bond acceptor (A) 1 Hydrogen-bond donor (D) 2 Hydrophobic group (H) 3 Negatively-charged atom (N) 4 Positively-charged atom (P) 5 Projected point (Q)—not used 6 Aromatic ring (R) 7 Custom (X) 8 Custom (Y) 9 Custom (Z)

Table B.5. Optional keywords for the Score Hypotheses step.

Keyword	Description
ALIGN_CUTOFF	Maximum RMS deviation in angstroms of aligned site points from two ligands, in angstroms (real). Default is 1.2 Å.
ALIGN_WEIGHT	Weighting factor of the site alignment term in the survival score (real). See Section 7.2.2 on page 70 for definitions. Default is 1.0.
BOXES_TO_KEEP	Percentage of top-scoring boxes to be retained for volume scoring after the first pass (integer). Default is 10.
CONFORMATION_PROPERTY	Name and weight of a conformation-dependent property to use in property scoring. Value contains property name and weight, separated by a comma. Multiple conformation property entries can be specified in the input file. For example, to use MMFF relative conformation energies and weight –0.1 the value of CONFORMATION_PROPERTY is r_mmod_Relative_Potential_Energy-MMFF94s,-0.1.
FEATURE_ALIGN_CUTOFF_FILE	Name of the file that contains feature-matching tolerances. See Section B.9 on page 187 for the format of this file.
MAX_BOXES	Maximum number of boxes to be scored. Default is 50. Overrides percentage specified by BOXES_TO_KEEP.
MIN_BOXES	Minimum number of boxes to be scored, equivalent to the minimum number of returned hypotheses per variant. Overrides percentage specified by BOXES_TO_KEEP. Default is 10.
PENALTY_CONST	Used to penalize hypotheses that do not have matches for all ligands. Default is 1.1. A value of 1.0 means that no penalty is applied.
PROPERTY_NAME	Name of the property to use in property scoring, e.g. r_m_phase_activity for the activity value. Only one property can be specified, and the property must be conformation-independent.
PROPERTY_WEIGHT	Weighting factor for the property (activity) term in the survival score. See Section 7.2.2 on page 70 for definitions. Default is 0.0.
SELECTIVITY_WEIGHT	Weighting factor for the selectivity term in the survival score. See Section 7.2.2 on page 70 for definitions. Default is 0.0.
USE_PROPERTY	Calculate property scores. Value can be true or false. Default is false.
USE_SELECTIVITY	Calculate selectivity score.Value can be true or false. Default is false.
USE_VOLUME	Calculate volume scores. Value can be true or false. Default is true.

Table B.5. Optional keywords for the Score Hypotheses step. (Continued)

Keyword	Description
VECTOR_CUTOFF	Minimum vector score value needed to keep the hypothesis. Default is 0.5.
VECTOR_WEIGHT	Weighting factor of the vector term in the survival score (real). See Section 7.2.2 on page 70 for definitions. Default is 1.0.
VOLUME_WEIGHT	Weighting factor of the volume term in the survival score (real). See Section 7.2.2 on page 70 for definitions. Default is 1.0.

An example of a Phase main input file is shown below. This example includes options for which defaults exist.

```
JOB_NAME=index_26
MIN_INTERSITE_DIST=2
NUM_SITES=5
FINAL_BOX_SIZE=2
MAXIMUM_DEPTH=5
MIN_NUM_LIGANDS_PER_BOX=5
MIN_MAX_SITES=0,0,5
MIN_MAX_SITES=1,0,5
MIN_MAX_SITES=2,0,5
MIN_MAX_SITES=3,0,5
MIN_MAX_SITES=4,0,5
MIN_MAX_SITES=6,0,5
ALIGN_CUTOFF=1.2
ALIGN_WEIGHT=1
VECTOR_CUTOFF=0.5
VECTOR_WEIGHT=1
VOLUME_WEIGHT=1
SELECTIVITY_WEIGHT=1
BOXES_TO_KEEP=100
PENALTY_CONST=1
MIN_BOXES=10
MAX_BOXES=50
LIGAND_DIR=ligands
BOXES_DIR=boxes
RESULT_DIR=results
LIGAND_NAME=120_ligand
LIGAND_NAME=121_ligand
LIGAND_NAME=130_ligand
LIGAND_NAME=132_ligand
LIGAND_NAME=BAM_ligand
LIGAND_NAME=BMZ_ligand
```

B.3 Feature Definition File

This file contains definitions used to specify pharmacophore features. The default feature definition file, `phase_feature.ini`, is provided in `$SCHRODINGER/phase-v`*version*`/data`. This file contains commonly used definitions for the six basic feature types. You can create your own feature definition file for a particular phase run. The file should be stored in the working directory for the run, and should be named *jobname*`_feature.ini`, where *jobname* is the name of the current job as specified in the Phase main input file.

Feature definition files contain blocks of data for each feature type. Feature types can be either the default types, such as acceptor, donor, or hydrophobic, or custom features. Each feature has a geometry, which can be one of `point`, `group`, or `vector`, and a projected point type, which depends on the geometry. Projected point types include donor and a range of acceptor types for vector geometries, and aromatic ring for group geometries.

Each block of data for a feature has the following format:

`#FEATURE`	Beginning of a new feature type block
`#IDENTIFIER` *char*	Single character feature identifier
`#COMMENT` *string*	Feature comments
`#INCLUDE`	Beginning of include block
pattern1	Pattern to include
pattern2	Pattern to include
...	
`#EXCLUDE`	Beginning of exclude block
pattern1	Pattern to exclude
pattern2	Pattern to exclude
...	

The `INCLUDE` block must contain at least one pattern; the `EXCLUDE` block can be empty. The identifier character must be one of the standard set, A, D, H, N, P, R, X, Y, or Z.

Individual patterns have the following format:

string1 string2 int1 int2 int3 int4 int5 [`#` *string3*]

The components of the patterns are described in Table B.6.

Table B.6. Feature definition pattern components.

Component	Description
string1	SMARTS pattern. For hydrophobic or aromatic features this string may be `default`, indicating that the default mechanism that calls underlying libraries should be used instead of pattern matching.
string2	Geometry definition. The allowed values are `point`, `vector`, and `group`. The `point` and `vector` strings may be followed by the index of an atom in the SMARTS pattern, in parentheses: for example, `point(2)`. This index defines the point or vector atom, and by default is the first atom in the SMARTS pattern. The `group` string may be followed by a comma-separated list of atom indices, in parentheses, which define the group atoms. The default is all atoms.
int1	Reserved for future use. Set it to 1.
int2	Reserved for future use. Set it to 1.
int3	Projected point type, which can be one of the following: 0 no projected points –1 donor –2 acceptor, sp^3, 1 lone pair (lp) –3 acceptor, sp^2, 1 lone pair –4 acceptor, sp, 1 lone pair –5 acceptor, sp^3, 2 lone pairs –6 acceptor, sp^2, 2 lone pairs –7 acceptor, sp, 3 lone pairs –8 aromatic ring –9 acceptor, planar, 3 lone pairs
int4	Indicates whether this pattern is used (0) or ignored (1).
int5	Indicates whether this pattern is a default pattern (1) or not (0).
string3	Optional comments.

An example of a feature definition file is shown below. This file contains definitions of 3 types: acceptor, donor and hydrophobic.

```
#FEATURE
#IDENTIFIER  D
#COMMENT  donor: hydrogen atom attached to oxygen, nitrogen, sulfur or carbon
#INCLUDE
[#1][O;X2]            vector(1)  0 1 -1   0  1 # OH
[#1]S[#6]             vector(1)  0 1 -1   0  1 # SH
[#1][#7]              vector(1)  0 1 -1   0  1 # any NH
#EXCLUDE
[#1]OC(=O)            point(1)   0 1  0   0  1 # exclude carboxyl group
[#1]O[S;X3]=O         point(1)   0 1  0   0  1 #
```

```
#FEATURE
#IDENTIFIER  A
#COMMENT  acceptor: oxygen, nitrogen or sulfur with at least one lone pair
#INCLUDE
n1c[nH]cc1            vector(1)  0  1 -3  0  1 # his
O=[C,c]               vector(1)  0  1 -6  0  1 # carbonyl oxygen
[O;X2](~[A,a])C       vector(1)  0  1 -5  0  1 # oxygen with two lone pairs
#EXCLUDE
O=C[O-,OH]            point      0  1  0  0  1 #
[#7;X3][*]=[O,S]      point      0  1  0  0  1 # general amide
[N;X3](C)(C)[C;X3]    point      0  1  0  0  1 #
[N;X3][a]             point      0  1  0  0  1 # planar N bonded to aring
#FEATURE
#IDENTIFIER  H
#COMMENT  hydrophobic feature
#INCLUDE
default               point      1  1  0  0  1 # default calls mmphob library
#EXCLUDE
```

In addition to features, projected point features can be included in the feature definition file. These features are defined by a point at a specified distance along the vector from a donor or acceptor atom. The format of a projected feature block is the same as for a feature, except that the initial keyword is `#PROJECTED_FEATURE`, and there is an additional `#EXTEND_DISTANCE` keyword that defines the distance of the projected point site from the donor or acceptor atom. An example of a projected point feature for a donor is given below.

```
#PROJECTED_FEATURE
#EXTEND_DISTANCE  1.8
#IDENTIFIER  D
#COMMENT  donor: hydrogen atom attached to oxygen, nitrogen, sulfur or carbon
#INCLUDE
[#1][O;X2]            vector(1)  0 1 -1   0  1 # OH
[#1]S[#6]             vector(1)  0 1 -1   0  1 # SH
[#1][#7]              vector(1)  0 1 -1   0  1 # any NH
#EXCLUDE
[#1]OC(=O)            point(1)   0 1  0   0  1 # exclude carboxyl group
[#1]O[S;X3]=O
```

B.4 Inactives Scoring Input File

The input file for inactives scoring (`phase_inactive`) contains *keyword=value* strings that provide instructions for scoring hypotheses with respect to inactives. An exclamation point "!" may be used to add comments to input file. The allowed keywords and their values are given in Table B.7. This file is automatically generated by the utility `pharm_score_inactives`. A sample input file is shown below.

Table B.7. Keywords for inactives scoring.

Keyword	Description
inactiveWeight	Required. Weight of the inactives score in the final score.
phaseOptionsFile	Required. Phase main input file for scoring inactives. The following keywords must be set in this file: FINAL_BOX_SIZE From phase_partition job. USE_VOLUME From phase_scoring job. ALIGN_CUTOFF From phase_scoring job. ALIGN_WEIGHT From phase_scoring job. VECTOR_WEIGHT From phase_scoring job. VOLUME_WEIGHT From phase_scoring job. LIGAND_NAME For each inactive molecule.
ligandArchive	Required. Name of the archive file (.tar) containing the ligand multi-conformer Maestro files for each ligand specified in the main input file. Must be stored in the current directory.
ligandDir	Required. Name of the directory used to store the ligands when they were archived, and therefore of the temporary directory that they will be extracted to.
hypoArchive	Required. Name of the archive file (.tar) containing the hypotheses. For each hypothesis, the files *hypoDir*/*hypoID*.mae, *hypoDir*/*hypoID*.tab and *hypoDir*/*hypoID*.xyz must be present in the archive. Must be stored in the current directory.
hypoDir	Required. Name of the directory used to store the hypotheses when they were archived, and therefore of the temporary directory that they will be extracted to.
survivalScore(*hypoID*)	Required. Survival score from actives scoring for the given hypothesis. There should be one record containing a survival score for each hypothesis for which inactive scores are wanted. Inactive scores are calculated only for hypotheses whose survival score is listed. Records may be deleted for hypotheses whose inactive score is not required.
featureFile	Required. Name of the feature definition file that was used to create the hypotheses.
tableFile	Required. Name of plain text file containing results.
csvFile	Name of CSV file containing results.

```
inactiveWeight=1
phaseOptionsFile=score_inactives_phase.inp
ligandArchive=score_inactives_ligandFiles.tar
ligandDir=.ligands.tmp
hypoArchive=score_inactives_hypoFiles.tar
hypoDir=.hypotheses.tmp
```

```
survivalScore(DHHRRR_37)=14.9862
survivalScore(DHHRRR_40)=14.8790
...
survivalScore(AAADHH_16)=13.4861
survivalScore(AAADHH_12)=13.4861
featureFile=score_inactives_feature.ini
tableFile=ScoreInactivesData.tab
csvFile=ScoreInactivesData.csv
```

B.5 Hypothesis Clustering Input File

The input file for clustering of hypotheses (`phase_hypoCluster`) contains *keyword*=*value* strings that provide instructions for clustering hypotheses according to their geometric similarity. An exclamation point "!" may be used to add comments to input file. The allowed keywords and their values are given in Table B.7. This file is automatically generated by the utility `pharm_cluster_hypotheses`. A sample input file is shown below.

Table B.8. Keywords for hypothesis clustering.

Keyword	Description
`phaseOptionsFile`	Required. Phase main input file with combined options from `phase_partition` and `phase_scoring`. The following options must be set in this file: `FINAL_BOX_SIZE` From `phase_partition` job. `ALIGN_CUTOFF` From `phase_scoring` job. `ALIGN_WEIGHT` From `phase_scoring` job. `VECTOR_WEIGHT` From `phase_scoring` job.
`hypoArchive`	Required. Name of the archive file (`.tar`) containing the hypotheses. For each hypothesis, the files *hypoDir*/*hypoID*`.mae`, *hypoDir*/*hypoID*`.tab` and *hypoDir*/*hypoID*`.xyz` must be present in the archive. Must be stored in the current directory.
`hypoDir`	Required. Name of the directory used to store the hypotheses when they were archived, and therefore of the temporary directory that they will be extracted to.
`featureDefFile`	Required. Name of the feature definition file used to create the hypotheses.
`featureTolFile`	Name of the feature-matching tolerances file. Required only if feature-matching tolerances were used to create the hypotheses
`clusterFile`	Required. Name of the output file containing results of the cluster analysis. Usually named *jobname*`_hypoCluster.clu`.

Table B.8. Keywords for hypothesis clustering.

Keyword	Description
linkageMethod	Method to be used for linking clusters. Allowed values are single, average, and complete. single — Use the highest similarity between any two objects from the two clusters. Produces diffuse, elongated clusters. average — Use the average similarity between all pairs of objects from the two clusters. complete — Use the lowest similarity between any two objects from the two clusters. Produces compact, spherical clusters.
survivalScore(*hypoID*)	Required. Survival score from scoring of actives for the given hypothesis. There should be one record containing a survival score for each hypothesis that is to be clustered. Records may be deleted for hypotheses that you do not wish to cluster.

```
phaseOptionsFile=cluster_hypotheses_phase.inp
hypoArchive=cluster_hypotheses_hypoFiles.tar
hypoDir=.hypotheses.tmp
featureDefFile=cluster_hypotheses_feature.ini
linkageMethod=complete
clusterFile=cluster_hypotheses_hypoCluster.clu
survivalScore(DHHRRR_37)=14.9862
survivalScore(DHHRRR_40)=14.8790
survivalScore(DHHRRR_43)=14.8454
...
```

B.6 Multiple QSAR Model Input File

The input file for phase_multiQsar contains *keyword*=*value* strings that provide instructions for creation and use of the QSAR models. An exclamation point "!" may be used to add comments to input file. The allowed keywords and their values are given in Table B.10. A sample input file is shown below.

Table B.9. Keywords for building multiple QSAR models with phase_multiQsar.

Keyword	Description
modelType	Type of model to build. Allowed values are atom and pharm. Default: atom.
numTrain	Required. Number of molecules in the training set. The first *numTrain* ligands from *phaseOptionsFile* (see below) are assigned to the training set, and the rest are assigned to the test set.

Table B.9. Keywords for building multiple QSAR models with phase_multiQsar.

Keyword	Description
`gridSpacing`	Grid spacing, in angstroms. Must lie between 0.5 and 4.0. The recommended and default value is 1.0.
`actProperty`	Required. The name of the activity property exactly as it appears in the ligand Maestro files. Missing activities are set to zero.
`maxFactors`	Required. Maximum number of PLS factors to include in the QSAR model. Models are createdfor the full sequence of models with the number of factors set to 1,...,`maxFactors`.
`useVolumeGroups`	Consider only atoms of the same MacroModel type when computing volume score overlaps. This favors alignments that superimpose chemically similar atoms. Allowed values are `true` and `false`. Default: `false`.
`phaseOptionsFile`	Required. Phase main input file with combined options for `phase_partition` and `phase_scoring`. The following keywords must be set in this file: `FINAL_BOX_SIZE` From `phase_partition` job. `USE_VOLUME` From `phase_scoring` job. `ALIGN_CUTOFF` From `phase_scoring` job. `ALIGN_WEIGHT` From `phase_scoring` job. `VECTOR_WEIGHT` From `phase_scoring` job. `VOLUME_WEIGHT` From `phase_scoring` job. `LIGAND_NAME` For each ligand in the training and test sets. `LIGAND_NAME` records for the training set should come first.
`ligandArchive`	Required. Name of archive file (`.tar`) containing ligand multiconformer Maestro files for each ligand specified in *phaseOptionsFile*.
`ligandDir`	Required. Name of the directory used to store the ligands when they were archived, and therefore of the temporary directory that they will be extracted to.
`hypoArchive`	Required. Name of archive file (`.tar`) containing hypothesis files for each hypotheses for which `hypoID` is specified. For each hypothesis, the files *hypoDir*/*hypoID*`.mae`, *hypoDir*/*hypoID*`.tab` and *hypoDir*/*hypoID*`.xyz` must be present in the archive.
`hypoDir`	Required. Name of the directory used to store the hypotheses when they were archived, and therefore of the temporary directory that they will be extracted to.

Table B.9. Keywords for building multiple QSAR models with phase_multiQsar.

Keyword	Description
`hypoID`	Required. Hypothesis for which QSAR model is to be built. There should be one record containing this keyword for each hypothesis for which you want a QSAR model. This list need not contain every hypothesis in the hypotheses archive.
`featureFile`	Required. Name of feature definition file used to create hypotheses.
`featureCutoffFile`	Name of file that defines positional tolerances for matching different types of pharmacophore features during ligand alignment. See Section B.9 on page 187 for a description of the format. Positional tolerances are enforced only after applying a tolerance to the intersite distances. If omitted, matching is done purely on intersite distances.
`featureRadiusFile`	Required if `modelType=pharm`. Name of file that defines feature radii.The format of this file is identical to *featureCutoffFile*, but the distances need not be the same.
`tableFile`	Required. Name of plain text file containing a summary of each QSAR model.
`csvFile`	Name of CSV file containing a summary of each QSAR model.
`resultArchive`	Required. Name of archive file (`.tar`) containing QSAR model results.
`resultDir`	Required. Name of directory for storing QSAR model results, relative to the current working directory.

```
modelType=atom
numTrain=27
gridSpacing=1
maxFactors=3
actProperty=r_phase_Ligand_Activity
useVolumeGroups=false
phaseOptionsFile=build_qsar_phase.inp
ligandArchive=build_qsar_ligandFiles.tar
ligandDir=.ligands.tmp
hypoArchive=build_qsar_hypoFiles.tar
hypoDir=.hypotheses.tmp
hypoID=DHHRRR_37
hypoID=DHHRRR_40
hypoID=DHHRRR_43
...
hypoID=AAADRR_22
hypoID=AAADHH_16
hypoID=AAADHH_12
featureFile=build_qsar_feature.ini
```

```
tableFile=BuildQsarData.tab
csvFile=BuildQsarData.csv
resultDir=BuildQsarResults
resultArchive=build_qsar_results.tar
```

B.7 QSAR Model Input File

The input file for the QSAR module contains *keyword*=*value* strings that provide instructions for creation and use of the QSAR model. An exclamation point "!" may be used to add comments to input file. The allowed keywords and their values are given in Table B.10. A sample input file is shown below.

Table B.10. Keywords for the QSAR input file

Keyword	Description
`runMode`	Legal values are `train` and `test`. Indicates whether a new model will be created (`train`) or whether an existing model will be imported (`test`). Required.
`maeFile`	Maestro file containing the molecules of interest. Required if `runMode=train`.
`actFile`	Text file containing activity data for the structures in `maeFile`. There should be one activity value per line, with no extraneous data or characters. Either `actFile` or `actProperty` (but not both) must be specified if `runMode=train`.
`actProperty`	The name of the activity property exactly as it appears in `maeFile`. Missing activities are set to zero. Either `actFile` or `actProperty` (but not both) must be specified if `runMode=train`.
`pharmFile`	Used for creating or testing a pharmacophore-based model. This file contains coordinates of site points that have been aligned to a particular hypothesis, one set of points for each molecule in `maeFile`. The file may be obtained by running `phase_fileSearch` on a Maestro file containing the molecules of interest. Valid when `runMode=train` or `runMode=test`. If `runMode=test` and `modelFile` contains a pharmacophore-based model, then `pharmFile` must be specified.
`featureRadiusFile`	File that defines the size of pharmacophore features. Each line in the file should contain a 1-character feature type, followed by a radius in angstroms. These radii are used only in the model creation process. Valid only when `runMode=train` and `pharmFile` has been specified.
`modelFile`	QSAR model file. The model is exported if `runMode=train`; the model is imported if `runMode=test`. Required if `runMode=test`.
`outputFile`	File for ordinary output. The default is to write to standard output.

Table B.10. Keywords for the QSAR input file

Keyword	Description
`numTrain`	The number of molecules in `maeFile` that will be used to train the model. The default is all molecules. Use the `duplex` option to control which molecules are assigned to the training set. Valid only when `runMode=train`.
`duplex`	A non-negative integer value that controls how the training set molecules are selected. If `duplex=0`, the first `numTrain` molecules in `maeFile` are used. If `duplex>0`, the value is treated as a random seed to sample `numTrain` molecules from `maeFile`. The default is `duplex=0`. Valid only when `runMode=train`.
`gridSpacing`	The distance in angstroms between neighboring points in the 3D grid. The default is 1.0. Values may range from 0.5 to 4.0. Valid only when `runMode=train`.
`maxFactors`	The maximum number of PLS factors to include in the QSAR model. Statistics and predictions are ultimately accessible for the full sequence of models with the number of factors set to 1,...,`maxFactors`. Valid only when `runMode=train`, in which case `maxFactors` must be specified.
`printModel`	Boolean (`true` or `false`) indicating whether or not a summary of the model should be written to `outputFile`. The default is `printModel=false`.
`printPred`	Boolean indicating whether or not predicted activities should be written to `outputFile`. If `runMode=train`, the training set and test set predictions are written separately. The default is `printPred=false`. `printPred=true` is allowed only when `maeFile` has been specified.
`printBits`	Boolean indicating whether or not volume occupation bit strings should be written to *outputFile*. `printModel=true` produces the full list of volume elements and atom classes that define the bit set. If `runMode=train`, bit strings for the training set and test set molecules are written separately. `printBits=true` is allowed only when `maeFile` has been specified.

```
runMode=train
maeFile=steroids.mae            ! 31 molecules
actFile=steroids_act.txt        ! 31 activity values
modelFile=steroids_model.dat
numTrain=21
duplex=1234567                  ! Random split: 21 train / 10 test
gridSpacing=1.0
maxFactors=4
printModel=true
printPred=true
printBits=true
```

B.8 Feature Frequencies File

This file is used to set the mimimum and maximum allowed feature frequencies for common pharmacophore perception. It is named `FeatureFreq.tab`. Each line contains the letter code for the feature type, followed by the minimum and the maximum number of occurrences of that feature in any hypothesis. The example below shows the default frequencies.

```
################################################################################
#                                                                              #
# Feature frequency file.  Used to set mimimum and maximum allowed feature     #
# frequencies for common pharmacophore perception.   You may change these      #
# limits, but do not make any other modifications to this file.                #
#                                                                              #
################################################################################
A 0 4
D 0 4
H 0 4
N 0 4
P 0 4
Q 0 0
R 0 4
X 0 4
Y 0 4
Z 0 4
END_OF_FEATURE_DATA
```

For example, the text `A 0 4` indicates that each common pharmacophore will be restricted to contain between zero and four acceptors (inclusive). If you had some prior knowledge of the problem at hand, you could adjust these frequencies to narrow the focus in accordance with that knowledge. For example, suppose it has been established that all actives bind to a specific site on the receptor through a hydrogen bond, where the ligand acts as an acceptor. In that case, you have justification to require that each common pharmacophore contain at least one acceptor.

B.9 Feature-Matching Tolerances File

This file is used to set feature-matching tolerances. When searching for matches, this file should be named *hypoID*`.tol`, where *hypoID* is the hypothesis identifier used to define other hypothesis-related files. Although this file contains no hypothesis-specific information, the naming convention is required for the file to be used when searching for matches for a specific hypothesis. You must make a copy of it with the appropriate name for each hypothesis for which you want to use feature-matching tolerances.

The file contains one line for each feature type for which tolerances are to be used. Each line consists of a single character feature type and a tolerance value in angstroms, separated by a space. If a feature type is omitted, a default tolerance of 1.0 is used for that feature type. The feature type `?` can be used to define a default cutoff for any feature type not listed in the file. The following is a sample feature-matching tolerances file:

```
################################################################################
#                                                                              #
# Feature matching tolerances applied when hypotheses are scored with respect  #
# to actives.  You may change the tolerances, but do not make any other        #
# modifications to this file.  To completely disable the use of tolerances,    #
# remove the FEATURE_ALIGN_CUTOFF_FILE option from score_actives_phase.inp.    #
#                                                                              #
################################################################################
A 1
D 1
H 1.5
N 0.75
P 0.75
R 1.5
X 1
Y 1
Z 1
END_OF_FEATURE_DATA
```

B.10 Hypothesis-Specific Tolerances File

This file is used to set tolerances for the specific features of a hypothesis in a search for matches. The file must be named *hypoID*`.tol`, where *hypoID* is the hypothesis identifier used to name hypothesis-related files. It contains multiple lines, each consisting of a single character feature type and a tolerance value, separated by a space. The file must contain one line for each site in the hypothesis, and there is a one-to-one mapping of the tolerances to the sites in the hypothesis. To see how the hypothesis maps to the reference ligand, you can use the Edit Hypothesis panel in Maestro.

The following is a sample hypothesis-specific tolerances file:

```
D 1.50
H 2.00
H 1.50
R 2.00
R 1.50
```

B.11 Site Mask File

This file is used to determine whether specific sites are matched or not in a partial match. The file must be named *hypoID*`.mask`, where *hypoID* is the hypothesis identifier used to name hypothesis-related files. The file must contain one line for each site in the hypothesis, with a 1, a 0 or a –1 on each line. These numbers determine how the site is matched:

1: Site must be matched.
0: Site can be matched but need not be matched
–1: Site must not be matched.

For example, consider a hypothesis that contains the site types D, H, H, R, and R. To require that every partial match contains the donor site but not the second hydrophobic site, the corresponding site mask file should contain the following 5 lines:

```
1
0
-1
0
0
```

B.12 Database Search Input File

This file is named *jobname*`_dbsearch.inp` and is automatically generated by the utility `phasedb_findmatches`. The file contains keyword=value pairs, as for the other input files. You should not normally need to modify this file, but in case you do, the full description of the keywords and their dependencies are given in Table B.11. Keywords that you should not modify are marked in the table. In this file, you must supply only the keywords that are relevant to the selected run mode. If you leave other keywords in the file, they will be flagged as errors, and the job will not run. You can comment out optional keywords or keywords that are illegal for the run mode using an exclamation point (!).

Table B.11. Description of keywords in the database search input file.

Keyword	Description
`jobName`	Job name. Required, must match the file name. Do not modify.
`inputFile`	Input file name, set automatically from `jobName`. Do not modify.
`logFile`	Log file name, set automatically from `jobName`. Do not modify.
`outputFile`	Output file name, set automatically from `jobName`. Do not modify.
`runMode`	Run mode. Required. Allowed values are `fetch`, `find+fetch`, and `flex`.

Table B.11. Description of keywords in the database search input file. (Continued)

Keyword	Description
`hypoFile`	Hypothesis coordinate file. Required in `find+fetch` and `flex` mode, illegal in `fetch` mode. Do not modify.
`featureFile`	Hypothesis feature definition file. Required in `find+fetch` and `flex` mode, illegal in `fetch` mode. Do not modify.
`deltaDist`	Tolerance (Å) used to match intersite distances in the find step. While `deltaDist` may be changed, the recommended value is twice the final box size that was used in the Find Common Pharmacophores step when the hypothesis was developed. The default final box size is 1.0 Å. Illegal in `fetch` mode.
`minSites`	Minimum number of sites in the hypothesis that must be matched. Optional. The allowed values depend upon the number of sites in the hypothesis: If the hypothesis contains 3 or more sites, `minSites` must be at least 3. If the hypothesis contains 1 or 2 sites, `minSites` must be 1 or 2. The default is to match all sites in the hypothesis. Illegal in `fetch` mode.
`siteMaskFile`	File that contains a mask indicating which sites must match when `minSites` is less than the number of sites in the hypothesis. The file should contain one line for each site in the hypothesis. Each line should contain a "1" or "0", where "1" indicates that the site specified on the corresponding line in the hypothesis file (e.g. the `.xyz` file) must be matched. The default site mask is all zeros (i.e., any site can be missed). Illegal in `fetch` mode.
`deltaHypoFile`	File that contains cutoffs for matching individual features in the hypothesis. Each line in the file should contain a 1-character feature type ("A", "D", "H", etc.), followed by a numerical cutoff in angstroms .There should be one line for each feature in the hypothesis.If this keyword is used, each match is checked to see whether any aligned site deviates from the hypothesis by more than the cutoff associated with that feature. The match is eliminated if a cutoff is exceeded. The default is to keep all matches that satisfy the `deltaDist` criterion. Illegal in `fetch` mode.
`featureCutoffFile`	File that contains cutoffs for matching feature types. Each line in the file should contain a 1-character feature type ("A", "D", "H", etc.), followed by a numerical cutoff in angstroms .The feature type "?" may be used to define a default cutoff for any feature type not listed in the file. If this keyword is used, each match is checked to see whether any aligned site deviates from the hypothesis by more than the cutoff associated with that feature. The match is eliminated if a cutoff is exceeded. The default is to keep all matches that satisfy the `deltaDist` criterion. Illegal in `fetch` mode.
`computeVector`	Turns on calculation of `vectorScore` during the find step. This can be set to `false`, in which case the fitness score will contain no `vectorScore` contribution. Illegal in `fetch` mode.

Table B.11. Description of keywords in the database search input file. (Continued)

Keyword	Description
`computeVolume`	Turns on calculation of `volumeScore` during the find step. Analogous to `vectorScore`. Illegal in `fetch` mode.
`refCtFile`	File that contains the reference conformation. Required if vector or volume scores are to be computed. Illegal in `fetch` mode. Do not modify.
`refConfIndex`	Identifies the reference conformation. This is a zero-based index, so a value of 16 indicates that the reference conformation is the 17th structure. Required if vector or volume scores are to be computed. Illegal in `fetch` mode. Do not modify.
`dbPath`	Absolute path to the database directory. Required. Do not modify.
`dbName`	The name of the database. Required.
`dbFormat`	Database format. Allowed values are `std`, `std+gz`, `tar`, and `tar+gz`. Indicates whether the Maestro and sites files are stored in standard (expanded) or archived format, and whether they are compressed (with `gzip`). Required. Do not modify.
`dbSubset`	Full path to database subset file.
`createNewDBSites`	If this keyword is set to `true`, pharmacophore sites will be generated during the search using the hypothesis feature definitions. This is necessary if the database feature definitions are not identical to the hypothesis feature definitions. Required for `find+fetch` mode, illegal otherwise.
`matchFile`	Match file. Holds a lookup table that is used to rapidly retrieve hits in the fetch step. Required in `find+fetch` and `fetch` mode, illegal in `flex` mode. Do not modify.
`saveMatchFile`	Option to save the match file. It is strongly recommended that you save the match file. Illegal in `fetch` and `flex` modes.
`alignWeight`	Weight of alignment score in fitness function. Default is 1.0.
`alignCutoff`	Alignment cutoff in fitness function. Default is 1.2 Å.
`vectorWeight`	Weight of vector alignment score in fitness function. Default is 1.0.
`vectorCutoff`	Vector score cutoff for filtering matches. Must be in the range [-1, 1]. Hits whose volume score is lower than this value are discarded. Default is -1.
`volumeWeight`	Weight of volume score in fitness function. Default is 1.0.
`volumeCutoff`	Volume score cutoff for filtering matches. Must be in the range [0,1]. Hits whose volume score is lower than this value are discarded. Default is 0.
`qsarModelFile`	QSAR model file. Optional. If present, the QSAR model in the named file is used to predict activities for the hits. Illegal in `find` mode. Comment out or remove only.

Table B.11. Description of keywords in the database search input file. (Continued)

Keyword	Description
`exclVolFile`	Excluded volume file. Optional. If present, excluded volumes are applied to filter the hits. Comment out or remove only.
`hitFile`	Hit file, in Maestro format. Required; default is *jobname*`-hits.mae`. Holds the conformations that produced matches, sorted by decreasing fitness and aligned to the hypothesis. Do not modify.
`maxHits`	The maximum number of hits that will be written to `hitFile`. If you want to view the hits in Maestro, this number should not be very large.
`hitGroupSize`	Controls how hits are grouped in `hitFile`. If `hitGroupSize > 0`, hits from the same molecule are grouped together and sorted by decreasing fitness. Groups of hits from different molecules are ordered by the top fitness value in each group. The maximum number of hits from any molecule cannot exceed `hitGroupSize`. If `hitGroupSize=0`, hits appear strictly in order of decreasing fitness, with no grouping by molecule and with no limit (other than `maxHits`) on the number of hits from any single molecule.

B.13 Maestro File Search Input File

The input file for searching a Maestro file for matches to a hypothesis is named *jobname*`_fileSearch.inp` and contains keyword=value pairs, as do the other input files. The full description of the keywords and their dependencies are given in Table B.12. You can comment out optional keywords or keywords that depend on other keyword settings using an exclamation point (!).

Table B.12. Description of keywords in the Maestro file search input file.

Keyword	Description
`maeFile`	Maestro file to be searched. May contain multiple conformations per molecule. Successive structures are treated as conformations of a single molecule only if the titles match and the connectivity specifications are identical. If successive structures differ only in their stereochemistry, they should have different titles if they are to be treated as separate molecules.
`flex`	Boolean (`true` or `false`) indicating whether or not flexible searching should be done. If `flex=true`, conformations are generated as needed for each molecule in `maeFile`. Use the other flex options to control how conformations are generated. The default is `flex=false`.
`flexSearchMethod`	Conformational search method. Legal values are `rapid` and `thorough`. The default is `rapid`. Valid only when `flex=true`

Table B.12. Description of keywords in the Maestro file search input file. (Continued)

Keyword	Description
`flexMaxConfs`	Maximum number of conformations per molecule to generate. The default is 100. Valid only when `flex=true`.
`flexMaxRelEnergy`	Conformational energy window, in kJ/mol. The default is 41.84, i.e., 10.0 kcal/mol. Valid only when `flex=true`.
`hypoID`	Hypothesis ID. Required. This is the stem of the hypothesis file names (e.g. *hypoID*`.tab`, *hypoID*`.xyz`) and must include the path if these files are not located in the current directory.
`deltaDist`	Tolerance (Å) used to match intersite distances. The recommended value is twice the final box size that was used in the Find Common Pharmacophores step when the hypothesis was developed. The default final box size is 1.0 Å; the default value of `deltaDist` is 2.0 Å
`minSites`	Minimum number of sites in the hypothesis that must be matched. Optional. The default is to match all sites in the hypothesis. Must be greater than or equal to 3. Do not use this option if the hypothesis contains only 1 or 2 sites.
`useSiteMask`	Boolean for applying a site mask to partial matches. The default is `true` if the file *hypoID*`.mask` is present. Set to `false` to disable. See Section B.11 on page 189 for information on site masks.
`useFeatureCutoffs`	Boolean for applying feature-based tolerances to matches. The default is `true` if the file *hypoID*`.tol` is present. Set to `false` to disable.
`useQSARModel`	Boolean for applying a QSAR model for the hypthesis to the hits. The default is `true` if the file *hypoID*`.qsar` is present. Set to `false` to disable.
`useExclVol`	Boolean for applying excluded volume filtering to hits. The default is `true` if the file *hypoID*`.xvol` is present. Set to `false` to disable.
`alignWeight`	Weight of alignment score in fitness function. Default is 1.0.
`alignCutoff`	Alignment cutoff in fitness function. Default is 1.2 Å.
`alignPenalty`	Alignment penalty in fitness function. Default is 1.2 Å.
`vectorWeight`	Weight of vector alignment score in fitness function. Default is 1.0.
`volumeWeight`	Weight of volume score in fitness function. Default is 1.0.
`hitFile`	Hit file, in Maestro format. Required. Holds the conformers that produced matches, aligned to the hypothesis. For a given molecule, matches are sorted by decreasing fitness. Scores and predicted activities (if any) are written as properties.
`maxHitsPerMol`	Maximum number of hits per molecule that will be written to `hitFile`. Default is 1.

Conversion from Phase 1.0

This appendix provides information on conversion of projects and data from Phase 1.0 to Phase 2.0. For all command-line usage, the following applies:

- You no longer need to define the `PHARMA_EXEC` environment variable to run the pharmacophore model utilities. If you have been defining this variable in a file that is executed every time you log in (e.g., `.cshrc`, `.bashrc`), you should remove or comment out the definition.

C.1 Command-line Pharmacophore Model Development

In addition to the new location, the command-line utilities have been completely rewritten since Phase 1.0. While there are many similarities to the old tools, the syntax of most commands has changed slightly. The overhaul was done to improve the command line interface and to make it easier to add new functionality. The changes have the following consequences:

- You cannot convert a Phase 1.0 command-line project to a Phase 2.0 project. You must start again with the original multi-conformer Maestro file and create a new Phase 2.0 project. This is necessary because of changes to the way in which hydrophobic features are perceived, hence there is no guarantee of a one-to-one correspondence between pharmacophore sites in Phase 1.0 and Phase 2.0. As a result, all pharmacophore-specific data must be recreated, and all steps in the workflow must be rerun.
- In Phase 1.0, the `VARIANT_NAMES` records were not generated automatically. The presence of these records makes it possible to exercise specific control over which variants are processed, and it allows `phase_partition` jobs to be run on multiple processors, where each processor operates on one or more variants.

C.2 Command-line Database Management

The most important change in the command line tools is that the databases you create will be recognized by the Phase graphical interface, so you have the option of searching from the command line or from Maestro.

The following changes should be noted:

- It is no longer necessary to run utilities that modify the database from the root database directory.
- There is no `-big` option when creating a new database. Molecules are automatically grouped into blocks of 5000, with each block stored in separate directory under *dbName*`_ligands`.
- There is no longer a `-title` option, so a connectivity check is always done for consecutive structures with the same title. This prevents inadvertent combination of conformers from chemical entities with the same title but different connectivities (e.g., tautomers created by LigPrep).
- A new `-ignore` option is provided so that titles can be completely ignored when perceiving conformers. This option was introduced because some third-party software is known to write SD files with a unique title for each conformer.
- Conformers may now be generated after adding molecules to the database, and pharmacophore sites are created concurrently.

Glossary

Active compound—A compound that shows high affinity for the biological target. Synonymous with the term *ligand*.

Active set—The set of active compounds that is used to develop a pharmacophore model. This set does not necessarily include all active compounds.

Excluded volume—A region of space in a pharmacophore hypothesis that should not be occupied by any atom of an active compound.

Feature—see **Pharmacophore feature**

Hit—A structure in a 3D database that is found to contain an arrangement of site points that can be mapped to the pharmacophore hypothesis. A hit is not necessarily active, but it is presumed to have a greater than average probability of being active if it was retrieved using a valid hypothesis.

Hypothesis—see ***n*-Point pharmacophore hypothesis**

Inactive compound—A compound that shows little or no affinity for the biological target.

Intersite distance—The distance between any two site points in a pharmacophore.

Ligand—see **Active compound**

Negative compound—A compound that is inactive, yet highly similar in structure to one or more known actives. Some compounds are negative because they lack certain key pharmacophore features found in true actives. Other negatives may actually satisfy exactly the same pharmacophore hypotheses as the actives, but possess extraneous structural characteristics that prevent binding.

Pharmacophore feature—A characteristic of chemical structure that may facilitate a noncovalent interaction between a ligand and a biological target. Examples are hydrogen-bond acceptor ("A"), hydrogen-bond donor ("D"), hydrophobe ("H"), positive ionic center ("P"), negative ionic center ("N").

Pharmacophore site—The labeling and location of a particular pharmacophore feature within a molecule. For example, a hydrogen bond acceptor site could simply be a nitrogen atom which carries an available lone pair. A hydrophobic site might be a methyl carbon or the centroid of a phenyl ring. The term *site point* is often used interchangeably with pharmacophore site.

***n*-Point pharmacophore**—Any 3D arrangement of *n* pharmacophore features.

***n*-Point pharmacophore hypothesis**—A specific 3D arrangement of *n* pharmacophore features, with associated uncertainties in the feature positions. High affinity ligands in their active conformations are expected to contain pharmacophore sites that can be mapped (within the limits of uncertainty) to any valid hypothesis. A given hypothesis may contain features that are associated with a single mode of binding, or it may contain features that are common to two or more modes of binding.

Reference ligand—The ligand that provides the pharmacophore that defines a hypothesis. In pharmacophore model development, this pharmacophore yields the highest multi-ligand alignment score for the active-set ligands. The reference ligand matches the hypothesis exactly, and has a perfect fitness score.

Site point—see **Pharmacophore site**

3D Database—A set of molecules, each of which is represented by one or more 3D conformational models, augmented with a pharmacophore-based representation of the molecules. A 3D database includes feature types and site point coordinates for each conformation.

Variant—The set of feature types in a pharmacophore. For example, the variant AHH indicates a 3-point pharmacophore containing one hydrogen bond acceptor and two hydrophobic sites.

Vector feature—A pharmacophore feature that contains directionality, such as a hydrogen bond acceptor, hydrogen bond donor, or aromatic ring. A vector feature does not necessarily have vector geometry.

Vector geometry—the geometric characteristics of hydrogen bond acceptors and donors. Refers to the direction of lone pairs in a hydrogen-bond acceptor or the direction of the heavy-atom–hydrogen-atom bond in hydrogen-bond donors. Features with vector geometry must be vector features.

Index

A

B

C

D

V

W

120 West 45th Street
32nd Floor
New York, NY 10036

101 SW Main Street
Suite 1300
Portland, OR 97204

3655 Nobel Drive
Suite 430
San Diego, CA 92122

Dynamostraβe 13
68165 Mannheim
Germany

QuatroHouse, Frimley Road
Camberley GU16 7ER
United Kingdom

SCHRÖDINGER®